การพยากรณ์สถานการณ์ covid-19 โดยใช้การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์

พีรพล คำพันธ์

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2567

การพยากรณ์สถานการณ์ covid-19 โดยใช้การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์

พีรพล คำพันธ์

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2567

วิทยานิพนธ์ เรื่อง "การพยากรณ์สถานการณ์ covid-19 โดยใช้การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์"

ของ พีรพล คำพันธ์

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบวิทยานิพนธ์

(ดร.ชูชาติ หฤไชยะศักดิ์)

..... ประธานที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร.จิรารัตน์ เอี่ยมสอาด)

..... กรรมการผู้ทรงคุณวุฒิภายใน

(รองศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

..... กรรมการผู้ทรงคุณวุฒิภายใน

(รองศาสตราจารย์ ดร.พนัส นัถฤทธิ์)

..... กรรมการผู้ทรงคุณวุฒิภายใน

(ดร.เศรษฐา ตั้งค้าวานิช)

อนุมัติ

.....

(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง การพยากรณ์สถานการณ์ covid-19 โดยใช้การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์
ผู้วิจัย พีรพล คำพันธ์
ประธานที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.จิรารัตน์ เอี่ยมสอาด
ประเภทสารนิพนธ์ วิทยานิพนธ์ ปร.ด. วิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2567
คำสำคัญ โควิด-19, Long Short-Term Memory, Multilayer Perceptron

บทคัดย่อ

ภายหลังการแพร่ระบาดของโควิด-19 ประเทศไทยได้รับผลกระทบหลายประการ โดยที่เห็นได้ชัดเจนที่สุดคือภาวะเศรษฐกิจตกต่ำและส่งผลกระทบต่อสุขภาพอย่างมาก รวมถึงการสูญเสียทรัพยากรทางการแพทย์และบุคลากรในการต่อสู้กับโรคระบาด อย่างไรก็ตาม ประเทศไทยยังขาดเครื่องมือวิเคราะห์และคาดการณ์ที่จำเป็นในการเตรียมพร้อมสำหรับสถานการณ์การแพร่ระบาดในอนาคต ดังนั้นผู้วิจัยจึงนำเสนอการพัฒนาแบบจำลองสำหรับการทำนายการแพร่กระจายของ COVID-19 โดยเฉพาะอย่างยิ่งการประยุกต์ใช้แบบจำลอง Long Short-Term Memory (LSTM) และแบบจำลอง Multilayer Perceptron (MLP) มาประยุกต์ใช้ในการสร้างแบบจำลองสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 ประเทศไทยมีทั้งหมด 77 จังหวัด ข้อมูลที่ใช้ในการทดลองนี้ได้มาจากกรมควบคุมโรคของรัฐบาลไทย การสร้างแบบจำลองใช้ข้อมูลสองประเภทคือ Dynamic-time Series (อนุกรมเวลา) และ Static Data (คงที่) มีสองขั้นตอนคือ 1) ใช้ LSTM เพื่อจัดการข้อมูลอนุกรมเวลาและ 2) ใช้โมเดล MLP เพื่อจัดการข้อมูลแบบคงที่ งานวิจัยนี้ทำนายจำนวนผู้ติดเชื้อสะสมและทำนายจำนวนผู้เสียชีวิตสะสม จากผลการทดลองพบว่าการใช้แบบจำลอง LSTM ร่วมกับ MLP เป็นแบบจำลองที่มีความถูกต้องมากที่สุดโดยความแม่นยำในการทำนายจำนวนผู้ติดเชื้อสะสมร้อยละ 98.60 และมีความแม่นยำในการทำนายจำนวนผู้เสียชีวิตสะสมร้อยละ 94.94 โดยการวัดค่าความคลาดเคลื่อนของโมเดลจาก Mean Absolute Error (MAE) นอกจากนี้ผลการทำนายของแต่ละจังหวัดในประเทศไทยที่ได้จากการพัฒนาแบบจำลองจะนำไปแสดงผลบนแผนที่ประเทศไทยบนเว็บแอปพลิเคชัน

Title THE PREDICTION OF COVID-19 SITUATION USING DEEP LEARNING AND GEOGRAPHIC INFORMATION SYSTEM
Author Peerapol Kompunt
Advisor Assistant Professor Jirarat Ieamsa-ard, Ph.D.
Academic Paper Ph.D. Dissertation in Computer Engineering - (Type 2.2), Naresuan University, 2024
Keywords COVID-19, Long Short-Term Memory, Multilayer Perceptron

ABSTRACT

Following the COVID-19 pandemic, Thailand has been affected in various ways, with the most noticeable impact being a significant economic downturn and a substantial impact on public health. This includes the loss of medical resources and personnel in the fight against the pandemic. Even having experienced these major social, economic and public health impacts, Thailand still lacks the necessary tools for analyzing and predicting future pandemic scenarios. Therefore, we propose a development model for predicting the impact of pandemics such as COVID-19, particularly by applying Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP) models to build a predictive model for forecasting the effects on public health of such events in Thailand. Thailand has 77 provinces, and the data used for this experiment was obtained from the Thai government's Department of Disease Control which collected the health data from all provinces. Two types of data were used in this experiment: dynamic (time series) and static. The process involves two steps: 1) using LSTM to handle time series data; and 2) using an MLP model to manage static data. This paper divides the prediction into two sets: predicting cumulative infections and predicting cumulative deaths. The experiment results showed that using a combination of the LSTM model with MLP provides the most accurate predictions with a prediction accuracy rate of 98.60% for cumulative cases and 94.94% accuracy for predicting cumulative deaths. Model accuracy was measured by using the mean absolute error (MAE). As part of the research, a web application showing the results graphically was also developed. This web application can display the health data for

each province on a map of Thailand.

ประกาศคุณูปการ

ข้าพเจ้าขอแสดงความขอบคุณอย่างสุดซึ้งต่อความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร.จิรารัตน์ เอี่ยมสอาด ที่ได้ให้การสนับสนุนอย่างเต็มที่ในฐานะประธานที่ปรึกษาวิทยานิพนธ์ ท่านได้มอบคำแนะนำและการช่วยเหลือที่มีคุณค่าอย่างยิ่งตลอดกระบวนการดำเนินการวิทยานิพนธ์ ซึ่งเป็นสิ่งที่มีความสำคัญต่อความสำเร็จของวิทยานิพนธ์ฉบับนี้

ข้าพเจ้าขอขอบพระคุณเป็นพิเศษต่อคณะกรรมการสอบวิทยานิพนธ์ ประกอบด้วย ดร.ชูชาติ หฤไชยะศักดิ์ ในตำแหน่งประธานกรรมการสอบวิทยานิพนธ์ ที่ได้ให้ข้อเสนอแนะอันเป็นประโยชน์ในระหว่างการสอบ อีกทั้ง ขอขอบพระคุณอย่างสูงต่อ รองศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล รองศาสตราจารย์ ดร.พนัส นัถฤทธิ์ และ ดร.เศรษฐา ตั้งค้าวานิช ที่ได้ร่วมเป็นกรรมการผู้ทรงคุณวุฒิภายในโดยได้ให้การพิจารณาและข้อเสนอแนะที่มีคุณค่า ซึ่งช่วยให้วิทยานิพนธ์ฉบับนี้มีคุณภาพและความสมบูรณ์ยิ่งขึ้น

ข้าพเจ้าขอขอบคุณ ศาสตราจารย์ ดร.ไพศาล มุณีสว่าง ซึ่งเป็นประธานที่ปรึกษาคนแรกที่ได้ให้ความรู้และแนวทางตั้งแต่วันแรกที่ข้าพเจ้าเริ่มศึกษา อีกทั้ง ข้าพเจ้าขอขอบคุณ มหาวิทยาลัยขอนแก่น และ ศาสตราจารย์ ดร.จักรชัย โสอินทร์ ที่ได้มอบโอกาสและทุนสนับสนุนการวิจัย ซึ่งมีบทบาทสำคัญในการดำเนินการวิทยานิพนธ์ฉบับนี้ให้สำเร็จลุล่วง

ข้าพเจ้าขอขอบคุณ นางสาวสุกัญญา ผนึกทอง ที่ได้คอยให้การสนับสนุนและคำปรึกษาด้านเอกสารวิชาการและเจ้าหน้าที่ทุกท่านที่ได้ให้ความช่วยเหลือและสนับสนุนในด้านต่างๆตลอดระยะเวลาการทำวิทยานิพนธ์

สุดท้ายนี้ ข้าพเจ้าขอขอบคุณตัวเองที่ได้ทุ่มเทและมุ่งมั่นในการทำวิทยานิพนธ์ฉบับนี้ รวมถึงครอบครัวที่ให้การสนับสนุนและกำลังใจมาโดยตลอด เป็นแรงผลักดันสำคัญในการก้าวข้ามอุปสรรคต่างๆ จนประสบความสำเร็จในครั้งนี้

พีรพล คำพันธ์

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การแพร่ระบาดของโรคโควิด-19 ในปัจจุบันส่งผลกระทบไปทั่วโลกทั้งในด้านเศรษฐกิจ สาธารณสุข การคมนาคม การท่องเที่ยว การศึกษา และแม้แต่ชีวิตประจำวันของมนุษย์ จุดเริ่มต้นของการแพร่ระบาดของ โควิด-19 เกิดขึ้นในเดือนธันวาคม 2019 ในเมืองอู่ฮั่น [1] ที่รู้จักกันในฐานะเมืองหลวงของมณฑลหูเป่ย์ในสาธารณรัฐประชาชนจีน อู่ฮั่นเป็นเมืองที่มีประชากรมากที่สุดในภาคกลางของจีน โดยมีประชากรมากกว่า 19 ล้านคน [2] เมื่อวันที่ 30 ธันวาคม 2019 คณะกรรมการสุขภาพแห่งชาติของสาธารณรัฐประชาชนจีนได้ออกประกาศอย่างเป็นทางการเกี่ยวกับโรคปอดบวมที่ไม่ทราบสาเหตุและเนื่องจากอู่ฮั่นมีประชากรจำนวนมากอัตราการติดเชื้อจึงเพิ่มขึ้นอย่างรวดเร็วภายในช่วงเวลาอันสั้นพบการติดเชื้อและมีผู้เสียชีวิตจำนวนมาก โดยโรค SARS-CoV-2 หรือที่เรียกว่าโควิด-19 [3] นั้นติดต่อจากคนสู่คนโดยสามารถแพร่เชื้อทางอากาศได้ ตามรายงานของคณะกรรมการสุขภาพแห่งชาติของสาธารณรัฐประชาชนจีนและทั่วโลก องค์การอนามัยโลก (WHO) จัดลำดับอาการห้าอันดับแรกของผู้ติดเชื้อ โควิด-19 ได้แก่ มีไข้ ไอ เหนื่อย หายใจไม่อิ่ม และปวดศีรษะ [4] และในผู้ป่วยที่มีโรคประจำตัว เช่น โรคหัวใจและเบาหวาน อาจมีอาการอื่นร่วมด้วย

ตามสถิติการแพร่ระบาดทั่วโลกของโควิด-19 โดยองค์การอนามัยโลก [5] ประเทศที่มีผู้ป่วยสะสมสูงสุด 5 อันดับแรกของโลก ได้แก่ สหรัฐอเมริกา จีน อินเดีย ฝรั่งเศส และเยอรมนี สหรัฐอเมริกาเป็นประเทศที่มีผู้ป่วยสะสมมากที่สุดจำนวน 102,977,369 ราย และเสียชีวิตสะสมจำนวน 1,120,529 ราย ประเทศที่ติดเชื้อมากที่สุดในเอเชีย 5 อันดับแรก ได้แก่ จีน อินเดีย ญี่ปุ่น เกาหลีใต้ และตุรกี ซึ่งอยู่ในทวีปเดียวกับประเทศไทย ประเทศจีนเป็นประเทศที่มีผู้ป่วยสะสมมากที่สุดในเอเชีย โดยมีผู้ป่วยสะสมจำนวน 99,240,488 ราย และเสียชีวิตสะสมจำนวน 120,912 ราย และประเทศไทยอยู่ในอันดับที่ 13 ของประเทศที่มีจำนวนผู้ป่วยสะสมมากที่สุด และอันดับที่ 11 ของประเทศที่มีจำนวนผู้เสียชีวิตสะสมมากที่สุด

การระบาดของโรคโควิด-19 ในประเทศไทยเริ่มตั้งแต่วันที่ 12 มกราคม 2563 [6] โดยผู้ป่วยรายแรกได้รับการยืนยันว่าเป็นนักท่องเที่ยวชาวจีนที่เข้ามาในประเทศไทยภายใน 2 สัปดาห์ เมื่อวันที่ 31 มกราคม 2563 คนขับแท็กซี่ที่ไม่เคยมีประวัติเดินทางไปต่างประเทศแต่เคยมีประวัติขับแท็กซี่ให้บริการลูกค้าชาวจีนได้รับการยืนยันว่าติดเชื้อโควิด-19 ขณะนี้จำนวนผู้ติดเชื้อยังคงเพิ่มขึ้นโดยประเทศไทยประสบปัญหาการแพร่ระบาดของโควิด-19 ถึง 5 ระลอก การแพร่ระบาดของโควิด-19 ส่งผลกระทบต่อประเทศไทยหลายด้าน เช่น เศรษฐกิจ การท่องเที่ยว [7] ระบบสุขภาพ สภาพการจ้าง

งาน และสิ่งแวดล้อม ความรุนแรงของผลกระทบทางเศรษฐกิจส่วนใหญ่ขึ้นอยู่กับปัจจัย 2 ประการ ได้แก่ 1) ระยะเวลาของการจำกัดการเคลื่อนไหวของผู้คนและกิจกรรมทางเศรษฐกิจในประเทศ และ 2) ประสิทธิผลของการตอบสนองทางการเงินในช่วงวิกฤต การมุ่งเน้นไปที่การใช้จ่ายด้านสุขภาพเพื่อควบคุมการแพร่กระจายของโรคและสนับสนุนรายได้ให้กับครัวเรือนที่ได้รับผลกระทบจากการแพร่ระบาดมากที่สุดได้ลดโอกาสที่เศรษฐกิจจะถดถอย

ในระลอกแรกของการระบาด ผลกระทบของมาตรการที่ดำเนินการโดยรัฐบาลนั้นมีการออกมาตรการล็อกดาวน์ด้วยการใช้จ่ายของรัฐบาลเพื่อพยุงเศรษฐกิจ สำนักงานสภาพัฒนาการเศรษฐกิจและสังคมแห่งชาติ (สศช.) ประเมินว่าการระบาดของโควิด-19 กระทบต่อการจ้างงานแรงงานใน 9 ภาคการผลิตกว่า 8 ล้านคน ไม่นับรวมแรงงานหลายล้านคนในภาคเกษตรหรือต่อแรงงานในภาคการท่องเที่ยวและภาคบริการที่เกี่ยวข้อง เช่น ธุรกิจร้านอาหารและโรงแรมที่ได้รับผลกระทบอย่างรุนแรง [8] จากการระบาดครั้งใหญ่ของโควิด-19 มีความรุนแรงมากตั้งแต่เดือนเมษายน 2564 ธนาคารโลกประเมินว่าการเติบโตทางเศรษฐกิจของประเทศไทยในปี 2564 ลดลงจากที่คาดการณ์ไว้ร้อยละ 3.4 เป็นร้อยละ 2.2 นอกจากนี้ การจ้างงานได้รับผลกระทบอย่างรุนแรง โดยเฉพาะในภาคบริการและธุรกิจ [9]

แม้เวลาผ่านไป 2 ปีนับตั้งแต่ไวรัสโควิด-19 มาถึงประเทศไทย การเสียชีวิตยังคงเกิดขึ้นทุกวัน มีผู้เสียชีวิตทั่วประเทศ 32,755 ราย โดยมีผู้ป่วยรายใหม่ยืนยันรวม 4,718,029 ราย หรือ 997 รายต่อวัน (อัปเดตเมื่อ 1 มกราคม 2567) [10] ด้วยเหตุนี้ การตระหนักรู้เกี่ยวกับไวรัสจึงแพร่หลายอย่างรวดเร็ว และการตัดสินใจมุ่งเน้นไปที่การป้องกันโรค การเฝ้าระวัง และการวิเคราะห์ข้อมูล ในขั้นต้น ประเทศไทยยังขาดเครื่องมือรวมถึงโมเดลการเรียนรู้เชิงลึกสำหรับการวิเคราะห์หรือคาดการณ์แนวโน้มการแพร่ระบาดของโควิด-19

ดังนั้นในงานวิจัยนี้จึงเสนอเครื่องมือวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 ด้วยสถาปัตยกรรมใหม่เพื่อวิเคราะห์และคาดการณ์แนวโน้มของการแพร่เชื้อโควิด-19 โดยใช้เทคนิคการเรียนรู้เชิงลึกเพื่อประเมินข้อมูลทางสถิติที่เกี่ยวข้องกับโควิด-19 ในแต่ละจังหวัดจาก 77 จังหวัดในประเทศไทย เครื่องมือนี้เป็นการประยุกต์ใช้โมเดล Long Short-Term Memory (LSTM) [11] และ Multilayer Perceptron (MLP) [12] ผสานเข้าด้วยกันเพื่อประมวลผลข้อมูลประเภทสองประเภทที่แตกต่างกันคือ ข้อมูลแบบอนุกรมเวลาและข้อมูลแบบคงที่ โดยโครงสร้างของสถาปัตยกรรมที่ออกแบบและพัฒนาขึ้นในงานวิจัยนี้ประกอบไปด้วยโมดูลหลักสามโมดูล คือ การจัดการข้อมูล การพัฒนาโมเดลสำหรับการทำนาย และการสร้างเว็บแอปพลิเคชัน ดังแสดงในภาพ 19

1.2 จุดมุ่งหมายของการวิจัย

เพื่อสร้างเครื่องมือสำหรับใช้ในการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 โดยการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกร่วมเพื่อพัฒนาโมเดลและข้อมูลทางด้านภูมิศาสตร์สารสนเทศสำหรับการสร้างแผนที่ของประเทศเพื่อรองรับการแสดงผลการทำนายจากโมเดลโดยการใช้เว็บแอปพลิเคชั่น โดยสามารถแยกเป็นแต่ละข้อได้ดังนี้

1. เพื่อสร้างเครื่องมือสำหรับการวิเคราะห์สถาการณ์การแพร่ระบาดของเชื้อโควิด-19
2. เพื่อสร้างโมเดลสำหรับการทำนายสถาการณ์การแพร่ระบาดของเชื้อโควิด-19
3. เพื่อสร้างเว็บแอปพลิเคชั่นสำหรับการแสดงผลการทำนายสถาการณ์การแพร่ระบาดของเชื้อโควิด-19 บนแผนที่ของประเทศไทย

1.3 ขอบเขตการวิจัย

1. ใช้เทคนิคการเรียนรู้เชิงลึกในการพัฒนาโมเดลสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19
2. ใช้ชุดข้อมูลการแพร่ระบาดของเชื้อโควิด-19 ภายในประเทศไทย
3. สร้างแผนที่ประเทศไทยเพื่อรองรับผลการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 ที่ได้จากโมเดลที่ได้พัฒนาขึ้น

1.4 นิยามศัพท์เฉพาะ

การเรียนรู้เชิงลึก [13] หมายถึง การเรียนรู้เชิงลึกเป็นหนึ่งในเทคนิคและโมเดลการเรียนรู้เครื่องจักรที่มุ่งเน้นการสร้างโครงข่ายประสาทเทียม [14] (neural networks) ที่มีความลึกมากเพื่อการประมวลผลข้อมูลและการตรวจจับลักษณะซับซ้อน คำว่า "ลึก" ในการเรียนรู้เชิงลึก หมายถึงการมีจำนวนชั้น [15] (layers) ในโครงข่ายประสาทเทียมที่มากมายทำให้สามารถเรียนรู้ลักษณะและโครงสร้างของข้อมูลได้อย่างละเอียด

โควิด-19 [16] หมายถึง Coronavirus Disease 2019 ซึ่งเป็นโรคระบาดที่เกิดขึ้นในปี 2019 โดยมีจุดเริ่มต้นที่ประเทศจีนและแพร่ระบาดไปยังส่วนอื่นๆของโลก โรคนี้เกิดจากการติดเชื้อไวรัสโคโรนา (Coronavirus) ชนิดใหม่ที่ตั้งชื่อว่า "SARS-CoV-2" (Severe Acute Respiratory Syndrome Coronavirus 2)

Long Short-Term Memory (LSTM) หมายถึง โครงข่ายประสาทเทียมประเภทหนึ่งที่พัฒนามาเพื่อการประมวลผลข้อมูลแบบลำดับ (sequential data) [17] โดยเฉพาะข้อมูลที่มีลำดับยาวและมีความสำคัญในข้อมูลก่อนหน้า.

Multilayer Perceptron (MLP) [18] หมายถึง โครงสร้างของโครงข่ายประสาทเทียมที่มีลักษณะแบบหลายชั้น (multilayer) ประกอบด้วยชั้นนำเข้า [19] ชั้นซ่อน [20] และชั้นผลลัพธ์ [21] โดยชั้นซ่อนสามารถมีหลายชั้นได้และเป็นส่วนสำคัญในการทำนายและการจัดรูปแบบข้อมูลที่มีความซับซ้อนได้

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

บทนี้กล่าวถึงเทคโนโลยีการเรียนรู้เชิงลึกและการวิเคราะห์งานวิจัยที่เกี่ยวข้องกับการพยากรณ์สถานการณ์ เช่น การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์ โดยมุ่งเน้นที่การประยุกต์ใช้เทคโนโลยีเหล่านี้ในบริบทของการพยากรณ์สถานการณ์การแพร่ระบาดของโรค เพื่อสร้างความเข้าใจความรู้เชิงลึกและสนับสนุนการพัฒนางานวิจัยในบริบทเดียวกันต่อไป

2.1 โครงข่ายประสาทเทียมแบบวนกลับ

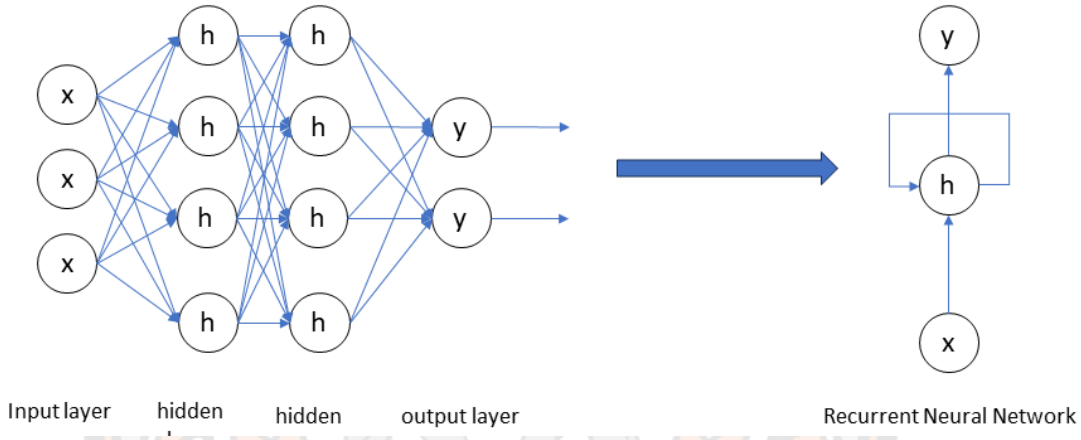
การใช้เทคโนโลยีการเรียนรู้เชิงลึก (Deep Learning) ในการทำนายอัตราการเพิ่มขึ้นของการติดเชื้อและการเสียชีวิตจากโควิด-19 ได้รับความนิยมเพิ่มขึ้นเนื่องจากความสามารถในการจัดการกับข้อมูลจำนวนมากและซับซ้อนเทคนิคการเรียนรู้เชิงลึกสามารถวิเคราะห์แนวโน้มและรูปแบบอนุกรมเวลา (Time-series Data) ได้อย่างมีประสิทธิภาพ ซึ่งช่วยในการคาดการณ์การเปลี่ยนแปลงในอัตราการติดเชื้อและการเสียชีวิตในอนาคต การใช้วิธีการเหล่านี้ทำให้สามารถจัดการกับสถานการณ์ที่ไม่แน่นอนและตอบสนองต่อการเปลี่ยนแปลงของการแพร่ระบาดได้อย่างรวดเร็วและแม่นยำ

การใช้การเรียนรู้เชิงลึก ในการทำนายอัตราการเพิ่มขึ้นของการติดเชื้อและการเสียชีวิตจากโควิด-19 ยังได้รับประโยชน์จากการรวมเทคนิค Geographic Information System (GIS) ซึ่งช่วยให้การวิเคราะห์ข้อมูลมีความละเอียดและแม่นยำมากขึ้น GIS สามารถจัดการกับข้อมูลเชิงพื้นที่และแสดงการแพร่ระบาดในรูปแบบแผนที่ที่ชัดเจน การรวมข้อมูลพื้นที่จาก GIS เข้ากับโมเดลการเรียนรู้เชิงลึก ช่วยให้สามารถติดตามและคาดการณ์แนวโน้มการแพร่ระบาดได้อย่างมีประสิทธิภาพ โดยการวิเคราะห์ข้อมูลตามภูมิภาคและพื้นที่เฉพาะ จึงเสริมสร้างความเข้าใจในแง่มุมภูมิศาสตร์ของการแพร่ระบาด และช่วยในการวางแผนการจัดการที่มีประสิทธิภาพมากขึ้น

โครงข่ายประสาทเทียมแบบวนกลับ [22] หรือ Recurrent Neural Network (RNN) เป็นโมเดลโครงข่ายประสาทเทียมที่ออกแบบมาเพื่อประมวลผลข้อมูลที่มีความเป็นลำดับ โดยโครงข่ายประสาทเทียมแบบวนกลับมีความสามารถในการจดจำและใช้ข้อมูลก่อนหน้า [23] การทำนายสถานะต่อไปเป็นสิ่งที่เครือข่ายประสาทแบบทั่วไปไม่สามารถทำได้ ส่วนใหญ่จะใช้ RNN สำหรับงานที่เกี่ยวข้องกับลำดับข้อมูล เช่น การประมวลผลภาษาธรรมชาติ [24] การแปลภาษา [25] การตรวจจับ [26] หรือใช้ในการทำนายด้านอื่นๆ [27, 28]

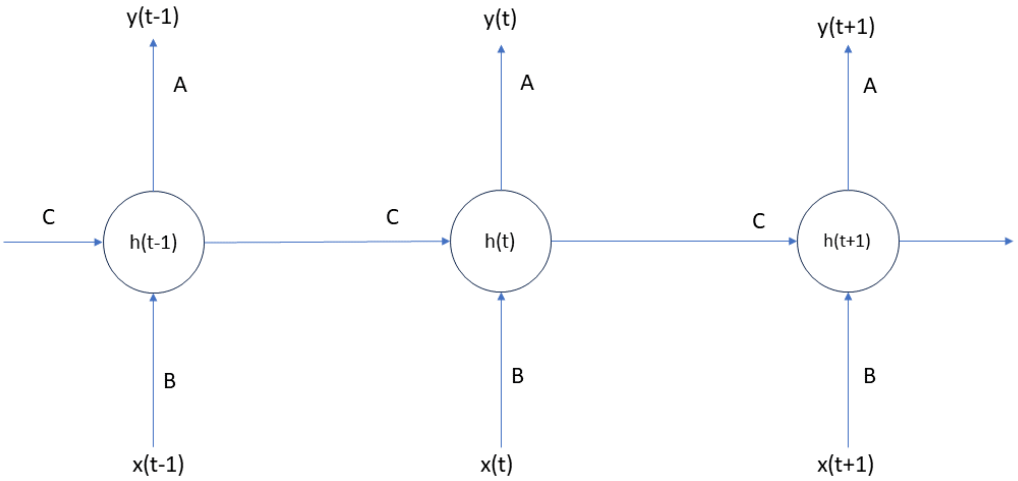
2.1.1 โครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับ

โครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับประกอบด้วยเซลล์ที่มีการเชื่อมต่อกันและมีความสามารถในการเก็บค่าสถานะก่อนหน้า [29] (Hidden State) และนำค่าสถานะนี้มาใช้ในการประมวลผลข้อมูลต่อไปในลำดับถัดไป การเชื่อมต่อระหว่างเซลล์ทำให้โครงข่ายประสาทเทียมแบบวนกลับสามารถจดจำข้อมูลที่เกิดขึ้นในอดีตและส่งต่อข้อมูลที่จำเป็นไปข้างหน้า [30] เนื่องจากโครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับมีความเกี่ยวข้องกับจำนวนลำดับเวลาที่แนวตรง [31] ดังภาพ 1



ภาพ 1 โครงสร้างโครงข่ายประสาทเทียมแบบวนกลับ

โครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับประกอบด้วยชั้นหลาย ๆชั้นที่เชื่อมต่อกันเป็นลำดับเวลาเพื่อประมวลผลข้อมูลตามลำดับที่เข้ามา ในแต่ละลำดับเวลาของโครงข่ายประสาทเทียมแบบวนกลับจะมีเซลล์ทำหน้าที่ในการคำนวณและเก็บค่าสถานะที่ซ่อนอยู่ที่ส่งผลต่อการประมวลผลข้อมูลในลำดับถัดไป [32] โดยมีรายละเอียดดังภาพ 2



ภาพ 2 โครงสร้างโครงข่ายประสาทเทียมแบบวนกลับแบบหลายชั้นที่เชื่อมต่อกันเป็นลำดับเวลา

เมื่อพิจารณาโครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับจากสมการที่ (1) จะได้ x คือชั้นข้อมูลนำเข้า (Input Layer) h คือชั้นซ่อน (Hidden Layer) และ y คือชั้นผลลัพธ์ (Output Layer) A B และ C คือพารามิเตอร์ของเครือข่ายที่ใช้เพื่อปรับปรุงผลลัพธ์ของโมเดล ในขณะใดขณะหนึ่ง T ข้อมูลนำเข้าปัจจุบันเป็นผสมผสานของข้อมูลนำเข้าที่ $x_{(t)}$ และ $x_{(t-1)}$ ผลลัพธ์ที่เกิดขึ้นในเวลาใดเวลาหนึ่งได้นำกลับเข้าสู่เครือข่ายเพื่อปรับปรุงผลลัพธ์ให้ดีขึ้น

$$h_{(t)} = f_c(h_{(t-1)}, x_{(t)}) \qquad (1)$$

2.1.2 โครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับแบบละเอียด

2.1.2.1 ข้อมูลเข้า (Input)

ในแต่ละลำดับเวลาข้อมูลที่เข้ามาจะส่งเข้าในเซลล์ของโครงข่ายประสาทเทียมแบบวนกลับเป็นเวคเตอร์ [33] ของข้อมูลที่เข้าในลำดับนั้น ๆ ข้อมูลในโครงข่ายประสาทเทียมแบบวนกลับคือชุดข้อมูลที่ใช้เป็นปัจจัยในการสร้างและประมวลผลโครงข่ายประสาทเทียมแบบ RNN ซึ่งเป็นโครงข่ายประสาทเทียมที่สามารถจดจำและใช้ข้อมูลที่เกิดขึ้นในขณะที่ประมวลผลข้อมูลก่อนหน้าได้ เพื่อให้ผลลัพธ์มีความสอดคล้องกับความเปลี่ยนแปลงของข้อมูลในลำดับเวลา ดังนั้นข้อมูลในโครงข่ายประสาทเทียมแบบวนกลับจะต้องมีลักษณะของลำดับของข้อมูลที่เกี่ยวข้องกับอนุกรมเวลา [34] ซึ่งสามารถอธิบายแบบละเอียดได้ดังนี้

ก) ลักษณะของข้อมูลในโครงข่ายประสาทเทียมแบบวนกลับ

เป็นข้อมูลที่ต้องการนำเข้าไปให้โมเดลประมวลผล เพื่อใช้ในการคำนวณหาผลลัพธ์หรือการทำนายต่อไป ข้อมูลเหล่านี้อาจเป็นตัวแปรหรือคุณสมบัติที่สำคัญในการตัดสินใจหรือสร้างการแบ่งประเภท [35] (Classification) หรือสร้างการพยากรณ์ (Forecasting) เช่น ข้อมูลการขายสินค้าในแต่ละวัน ราคาหุ้นในแต่ละวัน จำนวนผู้ใช้บริการในแต่ละช่วงเวลา สถิติการจราจรในแต่ละวัน ข้อมูลสภาพอากาศในแต่ละวัน เป็นต้น

ข) ลำดับของข้อมูลในโครงข่ายประสาทเทียมแบบวนกลับ

ลำดับของข้อมูลในโครงข่ายประสาทเทียมแบบวนกลับมีลักษณะเป็นลำดับของข้อมูลที่เกี่ยวข้องกับเวลาซึ่งแต่ละช่วงเวลาจะประกอบด้วยข้อมูลหลายๆค่า [36] เช่น ตัวแปร โดยที่ลำดับของข้อมูลเหล่านี้มีความสัมพันธ์ในช่วงหน้า-หลังกันและขึ้นอยู่กับความยาวของลำดับเวลาที่ต้องการใช้ในการทำนายหรือประมวลผล

2.1.2.2 ค่าสถานะที่ซ่อนอยู่ (Hidden State)

เซลล์ในแต่ละลำดับเวลาจะมีค่าสถานะที่ซ่อนอยู่ที่เป็นเวคเตอร์หรือเมทริกซ์ที่เก็บข้อมูลเกี่ยวกับความสำคัญของข้อมูลที่ผ่านมาในลำดับเวลา ค่าสถานะที่ซ่อนอยู่จะมีค่าเริ่มต้นที่กำหนดและจะปรับปรุงตามการประมวลผลข้อมูลแต่ละลำดับ [37] ค่าสถานะที่ซ่อนอยู่หรือเวคเตอร์ของสถานะที่ซ่อน [38] อยู่เป็นค่าหนึ่งที่คำนวณขึ้นมาในโครงข่ายประสาทเทียมแบบ RNN ที่ใช้ในการเก็บข้อมูลที่สำคัญในกระบวนการประมวลผลแต่ละลำดับของข้อมูลในลำดับเวลาและเป็นตัวแทนสำหรับสถานะที่แสดงถึงข้อมูลที่โมเดล การจำไว้ [39] จากลำดับก่อนหน้า ซึ่งส่งต่อไปให้กับลำดับถัดไปในกระบวนการประมวลผล นอกจากนี้ค่าสถานะที่ซ่อนอยู่ยังมีส่วนสำคัญในการทำนายหรือสร้างเอาต์พุตของโมเดลเช่นกัน [40] การทำงานของสถานะที่ซ่อนอยู่ในโครงข่ายประสาทเทียมแบบวนกลับสามารถอธิบายได้ดังนี้

ก) สร้าง Hidden State

เมื่อโมเดลโครงข่ายประสาทเทียมแบบวนกลับได้รับข้อมูลตัวแปรในลำดับเวลาปัจจุบัน (input) และค่าของสถานะที่ซ่อนอยู่ของลำดับก่อนหน้า (หรือค่าของสถานะที่ซ่อนอยู่ของครั้งแรกถ้าเป็นลำดับแรก) โมเดลจะนำข้อมูลนี้มาประมวลผลและคำนวณหาค่าของสถานะที่ซ่อนอยู่ของลำดับปัจจุบันใหม่ขึ้นมา โดยค่าของสถานะที่ซ่อนอยู่จะส่งต่อไปให้กับลำดับถัดไปในกระบวนการประมวลผล [41]

ข) เก็บข้อมูลที่สำคัญ

ค่าสถานะที่ซ่อนอยู่เป็นตัวแทนของข้อมูลที่สำคัญในลำดับก่อนหน้าและนำไปใช้ในการทำนายหรือประมวลผลข้อมูลในลำดับถัดไป ข้อมูลที่สำคัญที่จำไว้ในค่าสถานะที่ซ่อนอยู่รวมถึงสิ่งที่โมเดลได้เรียนรู้จากข้อมูลในลำดับก่อนหน้า

ค) ปรับความสัมพันธ์กับเวลา

ค่าสถานะที่ซ่อนอยู่จะทำให้โมเดลโครงข่ายประสาทเทียมแบบวนกลับมีความสัมพันธ์กับข้อมูลในลำดับเวลาที่ต่อเนื่องกันโดยสามารถจดจำและเรียนรู้ลำดับของข้อมูลในเวลาต่อเนื่อง [42] ไป ซึ่งเป็นคุณสมบัติที่ทำให้โครงข่ายประสาทเทียมแบบวนกลับสามารถทำนายหรือประมวลผลข้อมูลที่เกี่ยวข้องกับเวลาได้ [37]

2.1.2.2 ค่าสถานะที่ปรับปรุง (Updated Hidden State)

เพื่อความยืดหยุ่นในการปรับปรุงค่าสถานะที่ซ่อนอยู่ โครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับมีเมทริกซ์น้ำหนัก (Weight) [43] ที่ใช้ในการปรับปรุงค่าสถานะที่ซ่อนอยู่ซึ่งเมทริกซ์นี้จะคูณกับค่าสถานะที่ซ่อนอยู่และข้อมูลเข้าใหม่เพื่อให้เป็นค่าสถานะที่ปรับปรุงแล้ว การอัปเดตค่าสถานะที่ซ่อนอยู่ในสถานะที่ซ่อนอยู่มีขั้นตอนดังนี้

ก) ข้อมูลนำเข้า (Input Data)

เมื่อโมเดลโครงข่ายประสาทเทียมแบบวนกลับได้รับข้อมูลตัวแปรในลำดับเวลาปัจจุบัน (Input) และค่าของสถานะที่ซ่อนอยู่ของลำดับก่อนหน้า [44] (หรือค่าของสถานะที่ซ่อนอยู่ของครั้งแรกในกรณีที่เป็นลำดับแรก) โมเดลนำข้อมูลนี้มาประมวลผลและคำนวณหาค่าของสถานะที่ซ่อนอยู่ของลำดับปัจจุบันใหม่ขึ้นมา [45, 46] โดยค่าของสถานะที่ซ่อนอยู่จะส่งต่อไปให้กับลำดับถัดไปในกระบวนการประมวลผล

ข) Activation Function

หลังจากคำนวณหาผลลัพธ์ของาของสถานะที่ซ่อนอยู่แต่ละช่อง โมเดลใช้ฟังก์ชัน Activation Function เพื่อเพิ่มความหนักและความซับซ้อนในข้อมูล [44, 47, 48] ซึ่งสามารถช่วยให้โมเดลมีความสามารถในการจดจำและเรียนรู้ลำดับของข้อมูลได้

ค) อัปเดต Hidden State

หลังจากนั้น โมเดลจะใช้ผลลัพธ์ที่ได้จาก Activation Function เพื่ออัปเดตค่าาของสถานะที่ซ่อนอยู่ของลำดับปัจจุบัน โดยใช้ข้อมูลในลำดับก่อนหน้าและข้อมูลในลำดับปัจจุบันที่ประมวลผลไว้แล้วการอัปเดตาของสถานะที่ซ่อนอยู่นี้จะช่วยให้โมเดลสามารถเก็บข้อมูลที่สำคัญจากลำดับก่อนหน้าไว้เพื่อนำไปใช้ในลำดับถัดไป[44, 49]

ง) การทำซ้ำ (Iteration)

กระบวนการอัปเดตของสถานะที่ซ่อนอยู่จะทำซ้ำไปเรื่อยๆ [50] ตามจำนวนลำดับเวลาที่กำหนดในแต่ละลำดับเวลาโดยโมเดลนำข้อมูลในลำดับปัจจุบันมาประมวลผลเพื่อคำนวณหาของสถานะที่ซ่อนอยู่ของลำดับถัดไปซึ่งเป็นกระบวนการที่ทำให้โมเดลโมเดลโครงข่ายประสาทเทียมแบบวนกลับสามารถจดจำและเรียนรู้ข้อมูลที่เกี่ยวข้องกับเวลาได้

จ) ผลลัพธ์ (Output)

เซลล์ในแต่ละลำดับเวลาจะส่งผลลัพธ์ออกมาที่เป็นเวคเตอร์หรือเมทริกซ์ผลลัพธ์โดยผลลัพธ์นี้อาจนำไปใช้ในการทำนายหรือประมวลผลข้อมูลในขั้นตอนถัดไป ข้อมูลส่งออกของโมเดล RNN คือค่าที่ได้จากการทำนายหรือประมวลผลของโมเดลในลำดับเวลาปัจจุบัน โดยค่าผลลัพธ์นี้อาจเป็นตัวเลข หรือสัญลักษณ์ที่แทนคลาสหรือประเภทข้อมูลที่โมเดลพยายามทำนาย ขึ้นอยู่กับประเภทงานที่นำใช้ทำงานโครงข่ายประสาทเทียมแบบวนกลับมาประยุกต์ใช้ โดยในงานที่ใช้โครงข่ายประสาทเทียมแบบวนกลับในการทำนายค่าของผลลัพธ์ที่สำคัญคือการทำนายค่าในลำดับเวลาถัดไป ซึ่งอาจเป็นจำนวนตัวเลข เช่น จำนวนผู้ติดเชื้อในวันถัดไป เป็นต้น

ฉ) การเชื่อมต่อเวลา (Time-step Connection)

เซลล์ในลำดับเวลาถัดไปจะรับค่าสถานะที่ซ่อนอยู่จากเซลล์ในลำดับก่อนหน้าซึ่งช่วยให้โครงข่ายประสาทเทียมแบบวนกลับสามารถจดจำและใช้ข้อมูลที่ผ่านมาในการประมวลผลข้อมูลในลำดับถัดไปได้ Time-step Connection [51] หรือการเชื่อมต่อในแต่ละลำดับเวลาของโครงข่ายประสาทเทียมแบบวนกลับคือวิธีที่โมเดลโครงข่ายประสาทเทียมแบบวนกลับทำงานกับข้อมูลในลำดับเวลาต่อเนื่องกัน โดยให้ข้อมูลในลำดับเวลาก่อนหน้าเป็นอินพุตให้กับลำดับเวลาถัดไป เรียกว่าสัมพันธ์ในลำดับเวลา [52] (Temporal Relationship) หรือเชื่อมต่อในลำดับเวลา [53, 54] (Temporal Connection) ซึ่งเป็นคุณสมบัติที่ทำให้โครงข่ายประสาทเทียมแบบวนกลับเหมาะสำหรับการทำนายแบบที่มีความเป็นลำดับหรือข้อมูลที่เกี่ยวข้องกับเวลา

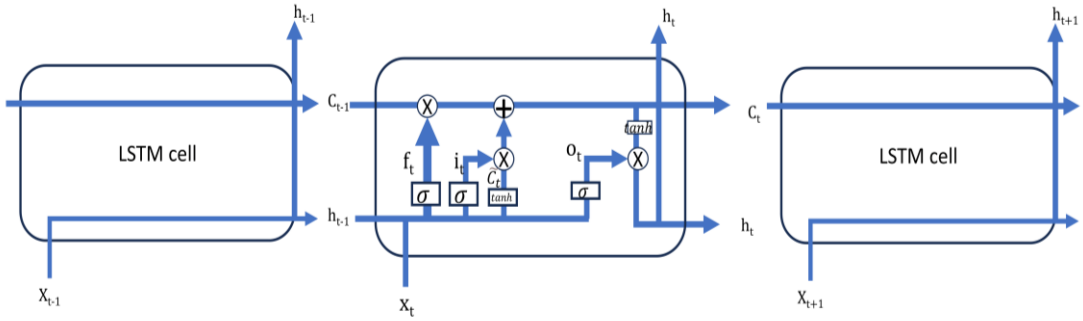
ในการทำนายโดยใช้ข้อมูลแบบลำดับเช่น การทำนายด้วยอนุกรมเวลา จำเป็นต้องให้โมเดลโครงข่ายประสาทเทียมแบบวนกลับเห็นความสัมพันธ์ระหว่างข้อมูลในลำดับเวลาต่างๆ ซึ่งได้สร้างขึ้นจากการเชื่อมต่อในแต่ละลำดับเวลา โดยในแต่ละขั้นตอนของการทำนาย ข้อมูลในลำดับเวลาก่อนหน้าจะส่งให้กับโครงข่ายประสาทเทียมแบบวนกลับเพื่อใช้ในการทำนายข้อมูลในลำดับเวลาถัดไป เมื่อโมเดลโครงข่ายประสาทเทียมแบบวนกลับทำนายข้อมูลในลำดับเวลาถัดไปแล้ว ข้อมูลที่ทำนายออกมาจะนำไปใช้ในขั้นตอนถัดไปเพื่อทำนายข้อมูลในลำดับเวลาถัดไปต่อไป และกระทำนี้ทำต่อเนื่องไปเรื่อยๆจนกว่าจะถึงลำดับเวลาสุดท้ายที่ต้องการทำนาย

เพื่อให้เกิดการเชื่อมต่อเวลา [55] ข้อมูลในแต่ละลำดับเวลาจะส่งเป็นอินพุตไปยังโมเดลเมื่อโมเดลโครงข่ายประสาทเทียมแบบวนกลับและข้อมูลที่ได้ทำนายในแต่ละลำดับเวลาก็จะนำไปใช้เป็น

อินพุตในการทำนายในลำดับเวลาถัดไปและขั้นตอนนี้จะเกิดขึ้นต่อเนื่องไปเรื่อยๆจนกว่าจะถึงลำดับเวลาสุดท้ายที่ต้องการทำนาย

2.2 Long Short-Term Memory (LSTM)

LSTM [56-61] เป็นโมเดลสามารถควบคุมค่าสถานะที่ซ่อนอยู่ในอดีตและทำงานได้ระยะยาว เนื่องจากมีประตูนำเข้า [30] (Input Gate) และประตูลืม[62] (Forget Gate) ที่ช่วยในการควบคุมข้อมูลในลำดับที่ผ่านมาและข้อมูลใหม่ที่เข้าสู่เซลล์ [63] การควบคุมความจำใน LSTM ช่วยให้โมเดลสามารถจดจำความสัมพันธ์ยาวนานของข้อมูลได้และเหมาะสำหรับการประยุกต์ในงานที่ต้องการการจดจำและการวิเคราะห์ข้อมูลในลำดับเวลาโดยมีโครงสร้างแบบง่าย [64] ดังแสดงในภาพ 3



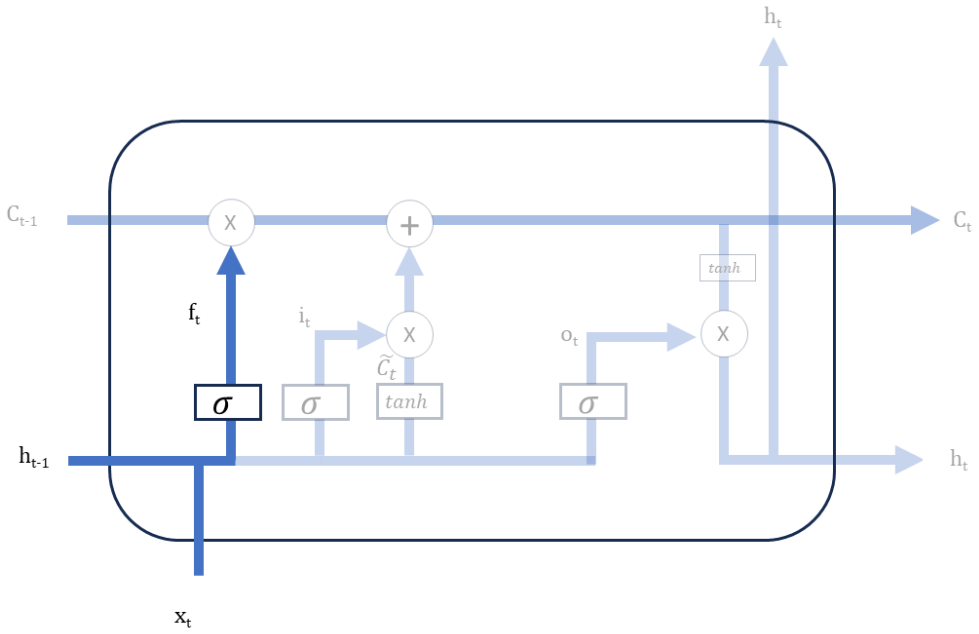
ภาพ 3 การเชื่อมต่อของโมเดล LSTM แบบหลายชั้น

ความแตกต่างพื้นฐานระหว่างโครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับและ LSTM คือโครงสร้างของชั้นซ่อนใน LSTM เป็นหน่วยหรือเซลล์ที่มีการควบคุมด้วยหน่วยของ Gated [65] หรือ Gated Cell [66, 67] ซึ่งประกอบด้วยสี่ชั้นที่มีความสัมพันธ์กันในลักษณะที่สร้างเอาท์พุตของเซลล์นั้นพร้อมกับสถานะเซลล์ โดยจะส่งต่อไปยังเลเยอร์ซ่อนถัดไป ตรงข้ามกับโครงสร้างของโครงข่ายประสาทเทียมแบบวนกลับที่มีเพียงชั้นเดียวของภายในที่เป็นชั้นของ TanH [68] เพียงอย่างเดียว

LSTM ประกอบด้วยประตูที่มีฟังก์ชัน Sigmoid [69] สามชั้นและชั้นของ TanH หนึ่งชั้น โดยประตูนำเข้ามาเพื่อจำกัดข้อมูลที่จะส่งผ่านเซลล์ [70] ซึ่งกำหนดว่าส่วนใดของข้อมูลจะนำมาใช้โดยเซลล์ถัดไปและส่วนใดควรตัดทิ้ง ผลลัพธ์ที่ได้จะอยู่ในช่วง 0-1 โดยทั่วไป '0' หมายถึง 'ปฏิเสธทั้งหมด' และ '1' หมายถึง 'รวมทั้งหมด'

โครงสร้างของโมเดล LSTM ประกอบไปด้วย ประตูลืม (Forget Gate) ประตูนำเข้า (Input Gate) เซลล์ความจำ (Memory Cell) ประตูการส่งออก (Output Gate) สถานะเซลล์ (Cell State) และสถานะที่ซ่อนอยู่ (Hidden State) มีรายละเอียดดังนี้

1) ประตูลืม (Forget Gate)



ภาพ 4 โครงสร้างการทำงานของประตูลืม

ประตูลืม [56] เป็นส่วนสำคัญที่ใช้ในการตัดสินใจว่าจะลืมหรือเก็บข้อมูลในอดีตที่สำคัญไว้ในเซลล์ความจำ (Memory Cell) [71] ประตูลืมจะคำนวณและส่งออกค่าเป็นเวคเตอร์ที่มีความยาวเท่ากับจำนวนข้อมูลในลำดับก่อนหน้า [72] โดยแต่ละองค์ประกอบของเวคเตอร์เป็นค่าระหว่าง 0 ถึง 1. เมื่อค่าเป็น 0 แสดงถึงการลืมข้อมูล และค่าเป็น 1 แสดงถึงการเก็บข้อมูลไว้ตามสมการที่ (2) โดยประตูลืมมีโครงสร้างดังภาพ 4

$$f_t = \sigma(w_f \bullet [h_{(t-1)}, x_t] + b_f) \quad (2)$$

โดยที่ f_t คือ ประตูลืม

σ คือ sigmoid ฟังก์ชัน

W_f คือ ค่าน้ำหนักของ matrices

h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ t-1)

x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t

b_f คือ ค่า bias

ดังนั้นประตูลืมช่วยให้ LSTM สามารถควบคุมการลืมข้อมูลที่ไม่สำคัญหรือไม่เกี่ยวข้องในอดีตได้และให้ความสำคัญกับข้อมูลที่สำคัญเพื่อใช้ในการทำนายและประมวลผลต่อไปได้อย่างมีประสิทธิภาพ ประตูลืมเป็นส่วนสำคัญที่ช่วยให้ LSTM สามารถจัดการและจดจำข้อมูลในลำดับเวลาที่ยาวนานได้

2) ประตูนำเข้า (Input Gate)

ประตูนำเข้า [73] เป็นส่วนที่ใช้ในการควบคุมข้อมูลใหม่ที่จะเข้าสู่เซลล์ความจำ ประตูนำเข้าใช้ฟังก์ชัน Sigmoid เพื่อทำการตัดสินใจว่าจะเก็บข้อมูลใหม่ [69] หรือไม่และใช้ฟังก์ชัน TanH ดังแสดงในภาพ 5 เพื่อสร้างเวคเตอร์ที่เก็บข้อมูลที่จะเพิ่มเข้าไปในเซลล์ความจำ ประกอบด้วยเวคเตอร์ที่มีความยาวเท่ากับจำนวนข้อมูลในลำดับก่อนหน้า แต่ละองค์ประกอบของเวคเตอร์แสดงถึงสัดส่วนของข้อมูลในลำดับก่อนหน้าที่ลืมหรือเก็บไว้ในเซลล์ความจำ ค่าฟังก์ชันอยู่ในช่วงระหว่าง 0 ถึง 1 โดยค่าเป็น 0 แสดงถึงการลืมข้อมูลและค่าเป็น 1 แสดงถึงการเก็บข้อมูลไว้ตามสมการที่ (3) และ (4)

$$i_t = \sigma(W_i \bullet [h_{(t-1)}, x_t] + b_i) \tag{3}$$

$$C_t = tanh \bullet (W_c \bullet [h_{(t-1)}, x_t] + b_c) \tag{4}$$

โดยที่ i_t คือ ประตูลืม

σ คือ sigmoid ฟังก์ชัน

C_t คือ ค่า candidate ของ cell state ที่เวลา t.

$\tanh$ คือ ฟังก์ชัน TanH

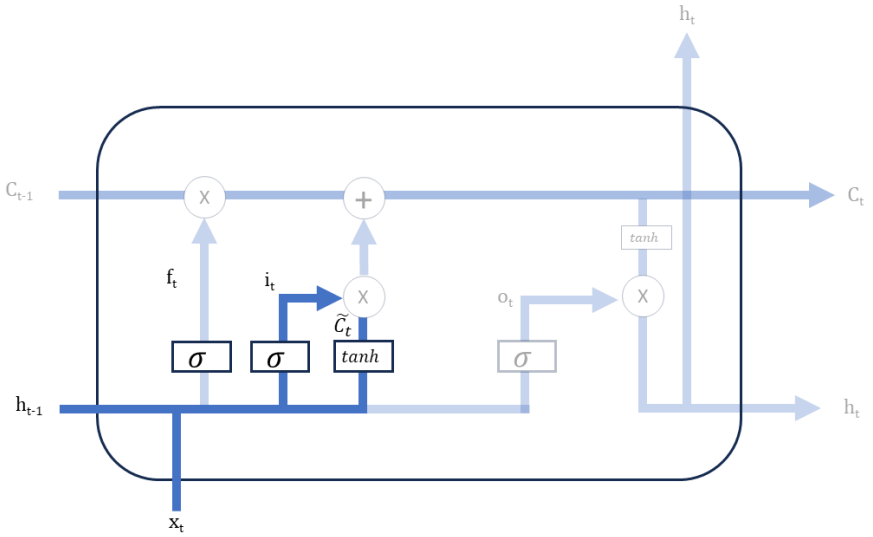
W_i, W_c คือ ค่าน้ำหนักของ matrices

h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ t-1)

x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t

b_i, b_c คือ ค่า bias

ประตูนำเข้าของ LSTM คือส่วนหนึ่งของเซลล์หน่วยควบคุมทางสถานะที่ใช้ในการควบคุมการเข้ารับข้อมูลเข้าสู่เซลล์หน่วยควบคุมทางสถานะ โครงสร้างของประตูนำเข้าแสดงดังภาพ 5 ซึ่งช่วยให้ LSTM สามารถเรียนรู้และจดจำข้อมูลที่เป็นลำดับเวลาได้ได้ดีกว่าโครงข่ายประสาทเทียมแบบวนกลับทั่วไป



ภาพ 5 โครงสร้างการทำงานของประตูนำเข้า

ในโมเดล LSTM เซลล์หน่วยควบคุมทางสถานะ (Cell State) เป็นส่วนที่ควบคุมการจำแนกข้อมูลในลำดับเวลา แต่เมื่อเซลล์ต้องการเลือกข้อมูลที่จะเก็บหรือลืมนั้น ใช้ประตูนำเข้าเพื่อช่วยในกระบวนการนี้สามารถควบคุมการนำเข้าข้อมูลใหม่เข้าสู่เซลล์หน่วยควบคุมทางสถานะ โดยประตูนำเข้าเป็นเส้นทางสำหรับเกิดการทำงานหากมีข้อมูลใหม่การทำงานของ Input Gate ใช้สองส่วนสำคัญคือ

ก) คำนวณ Gate Activation

LSTM ทำการคำนวณ Gate Activation โดยใช้ข้อมูลจากลำดับเวลาปัจจุบันและค่าสถานะที่ซ่อนอยู่จากลำดับเวลาก่อนหน้า [69] ซึ่งได้ค่า Activation ที่อยู่ในช่วง 0 ถึง 1 ซึ่งบ่งบอกถึงปริมาณข้อมูลที่เข้ามาที่ต้องการให้ผ่านเข้าสู่เซลล์ควบคุมทางสถานะ ค่า Activation ใกล้เคียง 0 จะแสดงถึงการตัดสินใจที่ต้องการให้ลืมข้อมูล และค่า Activation ใกล้เคียง 1 แสดงถึงการตัดสินใจที่ต้องการให้ข้อมูลเก็บไว้และนำไปใช้งานต่อ

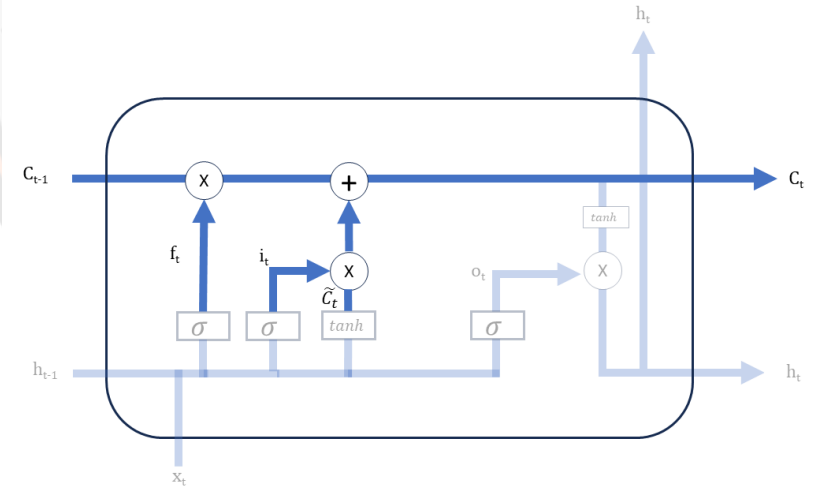
ข) การทำ Masking

หากมีค่า Activation ใกล้เคียง 0 (คือการตัดสินใจให้ลบข้อมูล) LSTM ทำการ Masking ข้อมูลนั้นออก คือ LSTM จะไม่นำข้อมูลนั้นไปใช้ในการอัปเดตค่าในเซลล์หน่วยควบคุมทางสถานะ ซึ่งการทำ Masking นี้ช่วยให้ LSTM สามารถลืมข้อมูลที่ไม่เกี่ยวข้องกับเวลาที่ผ่านไปได้

3) เซลล์ความจำ(Memory Cell)

เซลล์ความจำ [74, 75] ใน LSTM ประกอบด้วยเซลล์ความจำหลายตัว ที่เป็นแนวนอนและสามารถแยกควบคุมการเข้าถึงข้อมูลในลำดับเวลาที่ต่างกันได้ [76] ข้อมูลในเซลล์ความจำจะสั่งให้เกิดการบวกหรือลบกับข้อมูลในลำดับเวลาปัจจุบันตามความสำคัญของแต่ละส่วนที่ต้องการที่เก็บหรือลืม ดังแสดงในภาพ 6

เซลล์ความจำทำงานโดยการอ่านข้อมูลในลำดับเวลาก่อนหน้าและให้ความสำคัญให้กับข้อมูลที่ต้องการเก็บในเซลล์ความจำ ซึ่งการคำนวณความสำคัญนี้สามารถคำนวณได้โดยใช้สองประตูที่สำคัญในการทำงานของ LSTM คือประตูลืมและประตูนำเข้า



ภาพ 6 โครงสร้างการทำงานของเซลล์ความจำ

โดยการคำนวณของเซลล์ความจำนั้นใช้ประตูลืมและประตูนำเข้ามาเพื่อคำนวณในการตัดสินใจว่าจะเก็บหรือลืมข้อมูลโดยมีการดำเนินการของประตูลืมและประตูนำเข้าดังนี้

ประตูนำเข้า เป็นประตูที่คำนวณความสำคัญให้กับข้อมูลในลำดับเวลาปัจจุบันว่าจะต้องเก็บข้อมูลนั้นในเซลล์ความจำหรือไม่ ค่าของประตูนำเข้าอยู่ในช่วง [0, 1] โดยใช้ฟังก์ชัน Sigmoid เพื่อทำให้กำหนดค่าสำหรับการเก็บข้อมูลในลำดับเวลาปัจจุบัน

ประตูลืม เป็นประตูที่คำนวณความสำคัญให้กับข้อมูลในเซลล์ความจำของลำดับเวลาก่อนหน้าว่าจะต้องลบข้อมูลนั้นออกจากเซลล์ความจำหรือไม่ ค่าของประตูลืมจะอยู่ในช่วง [0, 1] โดยใช้ฟังก์ชัน Sigmoid เพื่อทำให้กำหนดค่าสำหรับการลืมข้อมูลในเซลล์ความจำของลำดับเวลาก่อนหน้า

เมื่อมีประตูลืมที่คำนวณความสำคัญให้กับข้อมูลในลำดับเวลาก่อนหน้าและข้อมูลในเซลล์ความจำต่อมาประตูความจำจะนำเอาข้อมูลในลำดับเวลาปัจจุบันที่ได้รับประตูนำเข้ามาบวกกับข้อมูลในเซลล์ความจำในลำดับเวลาก่อนหน้าที่ได้รับประตูลืมที่นำไปใช้ในการลบข้อมูล ทำให้ข้อมูลในเซลล์ความจำอัพเดตข้อมูลในลำดับเวลาปัจจุบันได้ [77] โดยสามารถอธิบายได้จากสมการที่ (5)

$$C_t = f_t \bullet C_{t-1} + i_t \bullet C_t \tag{5}$$

โดยที่ C_t คือ ประตูความจำ

C_t คือ ค่า candidate ของ cell state ที่เวลา t.

C_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ t-1)

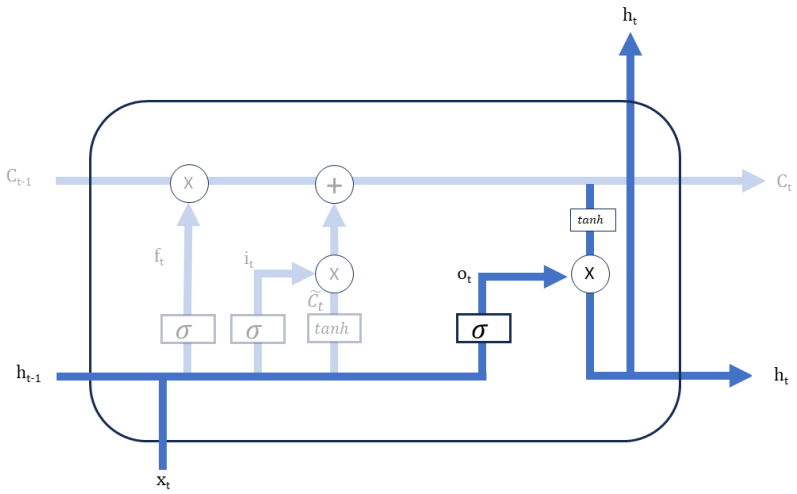
f_t, i_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t

4) ประตูการส่งออก (Output Gate)

ประตูการส่งออก [56, 78] เป็นส่วนที่ใช้ในการควบคุมผลลัพธ์ที่ออกจากเซลล์ความจำ ประตูการส่งออกใช้ฟังก์ชันสเต็ป [79, 80] sigmoid เพื่อทำการตัดสินใจว่าจะส่งผลลัพธ์ออกหรือไม่ และใช้ฟังก์ชัน TanH ดังแสดงใน ภาพ 7 เพื่อแปลงค่าสถานะที่ซ่อนอยู่ในลังความจำเป็นเป็นผลลัพธ์

ประตูการส่งออกใน LSTM เป็นส่วนที่ควบคุมการส่งผลลัพธ์ที่เกิดขึ้นจากประตูความจำหรือเซลล์ความจำในโมเดล หมายถึง LSTM โดยช่วยให้โมเดลสามารถควบคุมการแสดงผลข้อมูลที่เกิดขึ้นในประตูความจำและให้ความสำคัญในการคำนวณผลลัพธ์ที่จะส่งออกมาตามลำดับเวลาของข้อมูล

ประตูการส่งออกมีลักษณะคล้ายกับประตูนำเข้าโดยมีการคำนวณค่าความสำคัญที่อยู่ในช่วง [0 ถึง 1] ด้วยฟังก์ชัน Sigmoid ซึ่งค่าของประตูการส่งออกที่ใกล้ 1 หมายถึง LSTM จะส่งข้อมูลในประตูความจำออกมาเต็มที่ และค่าที่ใกล้ 0 หมายถึง LSTM จะไม่ส่งข้อมูลออกมาโดยโครงสร้างของประตูส่งออกแสดงดังภาพ 7



ภาพ 7 โครงสร้างการทำงานของประตูส่งออก

กระบวนการทำงานของประตูการส่งออกจะเกิดขึ้นหลังจากประตูความจำได้รับข้อมูลที่ผ่านการกรองด้วยประตูนำเข้าและประตูลืมโดยข้อมูลที่ส่งออกมาที่ประตูการส่งออก จากนั้นประตูการส่งออกจะนำค่าสำคัญที่คำนวณได้มาคูณกับผลลัพธ์ที่ได้จากประตูความจำเพื่อให้เกิดผลลัพธ์ที่สอดคล้องกับข้อมูลที่ LSTM ต้องการให้ส่งออกมา

ประตูการส่งออก ใน LSTM เป็นส่วนที่ควบคุมการส่งผลลัพธ์ที่เกิดขึ้นจากประตูความจำหรือเซลล์ความจำในโมเดล LSTM โดยช่วยให้โมเดลสามารถควบคุมการแสดงผลข้อมูลที่เกิดขึ้นในประตูความจำและให้ความสำคัญในการคำนวณผลลัพธ์ที่จะส่งออกมาตามลำดับเวลาของข้อมูล สำหรับขั้นตอนการคำนวณประตูการส่งออกใน LSTM สมการคำนวณค่าของประตูการส่งออก [77] ได้ดังสมการที่ (6)

$$o_t = \sigma(W_o \bullet [h_{(t-1)}, x_t] + b_o) \tag{6}$$

โดยที่ o_t คือ ประตูส่งออก

σ คือ sigmoid ฟังก์ชัน

W_0 คือ ค่าน้ำหนักของ matrices

h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ t-1)

x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t

b_0 คือ ค่า bias

5) สถานะเซลล์ (Cell State)

สถานะเซลล์ [81, 82] หรือ Cell State เป็นตัวแปรที่เก็บข้อมูลหลักของ LSTM ที่มีความสำคัญในการจดจำข้อมูลในระยะยาว ในแต่ละขั้นตอนของการทำนายข้อมูลใหม่จะเพิ่มเข้าไปในสถานะเซลล์และข้อมูลที่ไม่เกี่ยวข้องจะทำการลบออก

สถานะเซลล์ ใน LSTM เป็นตัวเก็บข้อมูลที่ใช้ในการจดจำและควบคุมข้อมูลหลักของเครือข่าย เป็นส่วนที่สำคัญที่ช่วยให้ LSTM สามารถจัดการกับความขึ้นอยู่ในระยะยาวได้ สถานะเซลล์คือส่วนสำคัญที่ทำให้ LSTM สามารถจดจำข้อมูลในช่วงเวลายาวนานและควบคุมการส่งผ่านข้อมูลที่สำคัญไปยังขั้นตอนถัดไป สถานะเซลล์เป็นเวคเตอร์ที่มีขนาดคงที่และเปลี่ยนแปลงเพียงบางส่วนของข้อมูลเท่านั้น ในแต่ละขั้นตอนของการทำนายสถานะเซลล์จะอัปเดตโดยใช้ประตูต่างๆ ของ LSTM ที่ควบคุมการเพิ่มหรือลบข้อมูลในสถานะเซลล์ กระบวนการทำงานของสถานะเซลใน LSTM สามารถอธิบายได้ดังนี้

ก) ลืมข้อมูลที่ไม่สำคัญ ในขั้นตอนนี้ ของ LSTM ใช้ประตูลืมเพื่อตัดสินใจว่าจะลืมข้อมูลใดบ้างในสถานะเซลล์เก่า C_{t-1} โดยปรับค่าสถานะเซลล์เก่าด้วยค่าของประตูลืม ข้อมูลที่ไม่สำคัญจะลบออกจากสถานะเซลล์เก่า ค่าของประตูลืมเป็นค่าระหว่าง 0 ถึง 1 โดยเพิ่มขึ้นจากข้อมูลเข้าสู่ประตูลืมแล้วผ่านฟังก์ชัน Sigmoid ที่ปรับค่าข้อมูลให้อยู่ระหว่าง 0 และ 1

ข) เพิ่มข้อมูลใหม่ ในขั้นตอนนี้โมเดลLSTM จะใช้ประตูนำเข้าเพื่อตัดสินใจว่าจะเพิ่มข้อมูลใดเข้าสู่สถานะเซลล์ใหม่ C_t โดยคำนวณการอัปเดตข้อมูลใหม่จากข้อมูลเข้า (x_t) และข้อมูลที่ผ่านการแปลงโดย TanH(C_t) ผ่านประตูนำเข้าโดยใช้ฟังก์ชัน sigmoid ประตูนำเข้าเป็นค่าระหว่าง 0 ถึง 1 ซึ่งจะเป็นตัวกำหนดว่าจะเพิ่มข้อมูลใดเข้าสู่สถานะเซลล์ใหม่

ค) อัปเดตสถานะเซลล์ใหม่ ในขั้นตอนนี้สถานะเซลล์ใหม่ C_t จะอัปเดตโดยการคูณสถานะเซลล์เก่า C_{t-1} ด้วยค่าประตูลืมที่ปรับแล้ว (f_t) เพื่อลบข้อมูลที่ไม่สำคัญออก จากนั้นจะบวกกับผลคูณระหว่างประตูลื่มที่ปรับแล้ว (i_t) และข้อมูลใหม่ที่ผ่านการแปลง (C_t) ผ่านประตูนำเข้า สุดท้ายจะได้สถานะเซลล์ใหม่ C_t ที่อัปเดตแล้ว

6) สถานะที่ซ่อนอยู่ Hidden State

สถานะที่ซ่อนอยู่ [83, 84] หรือ Hidden State ใน LSTM เป็นผลลัพธ์หรือข้อมูลที่ออกมาจากขั้นตอนปัจจุบันของ LSTM ซึ่งได้รับการปรับปรุงและแก้ไขจากสถานะเซลล์ สถานะที่ซ่อนอยู่เป็นผลลัพธ์ที่นำไปใช้ในขั้นตอนถัดไปของการทำนายหรือการประมวลผลข้อมูลในเครือข่ายเพื่อเพิ่มความเข้าใจและเพิ่มประสิทธิภาพในการแก้ไขปัญหาที่เกี่ยวข้องกับลำดับ

สถานะที่ซ่อนอยู่เป็นเวคเตอร์ที่มีขนาดคงที่และเปลี่ยนแปลงเพียงบางส่วนของข้อมูลเท่านั้น สำหรับแต่ละขั้นตอนในการทำนาย สถานะที่ซ่อนอยู่จะคำนวณโดยใช้ข้อมูลจากสถานะเซลล์และ

ข้อมูลนำเข้าใหม่และสถานะที่ซ่อนอยู่ในขั้นตอนก่อนหน้า การทำงานของ Hidden State ใน LSTM มีลักษณะเป็นดังนี้

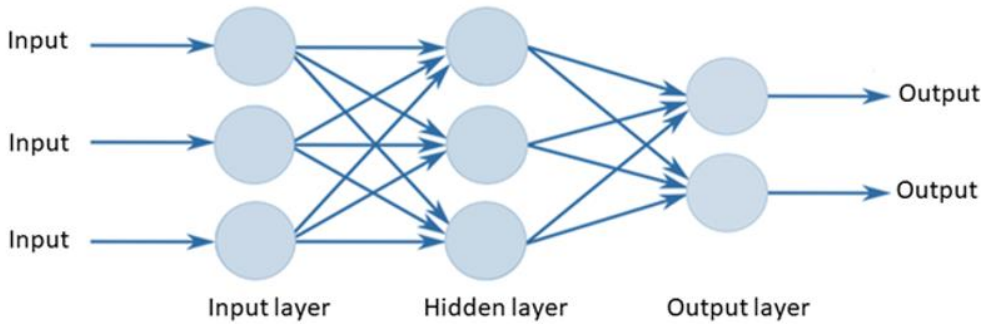
ก) การประมวลผลคำนวณ ทุกขั้นตอนของลำดับเวลา LSTM จะคำนวณผลลัพธ์อย่างน้อย 1 ครั้งเพื่อสร้างสถานะที่ซ่อนอยู่ใหม่โดยใช้ข้อมูลปัจจุบัน (x_t) และสถานะที่ซ่อนอยู่ก่อนหน้า (h_{t-1}) โดยป้อนข้อมูลเข้าสู่ชั้นต่างๆภายในโครงสร้าง LSTM

ข) การส่งต่อข้อมูล สถานะที่ซ่อนอยู่ใหม่ที่คำนวณขึ้นมาในแต่ละขั้นตอนจะนำไปใช้ในขั้นตอนถัดไปของลำดับเวลานั่นคือสถานะที่ซ่อนอยู่ของขั้นตอนก่อนหน้าจะเป็นข้อมูงเข้าของขั้นตอนถัดไปเพื่อคำนวณสถานะที่ซ่อนอยู่ใหม่

ค) การเก็บความสำคัญของข้อมูล สถานะที่ซ่อนอยู่มีความสำคัญเพราะเป็นการนำร่องข้อมูลที่ผ่านโครงสร้าง LSTM ไปยังส่วนอื่นๆของโมเดล เช่น ใช้ในการทำนายหรือการตัดสินใจ สถานะที่ซ่อนอยู่เป็นการเก็บความสามารถในการจดจำและแบ่งปันข้อมูลที่สร้างขึ้นจากข้อมูลที่ผ่านโครงสร้าง LSTM ทั้งในระยะยาวและระยะสั้น

2.3 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) [18, 85] เป็นหนึ่งในโมเดลเรียนรู้เชิงลึกที่ใช้ในการประมวลผลข้อมูล เป็นส่วนหนึ่งของกลุ่มของ Artificial Neural Networks (ANNs) [86] หรือเรียกว่าโครงข่ายปราสาทเทียม ซึ่งจำลองการทำงานของระบบประสาทของมนุษย์ในการประมวลผลข้อมูลและการตัดสินใจ MLP นิยมใช้กันในหลายด้าน เช่น การจำแนกประเภทของข้อมูล [87], การทำนายค่า [88] (Regression) และงานที่เกี่ยวกับการทำนายความรู้สึก [89] ความสามารถของ MLP จะขึ้นอยู่กับการกำหนดโครงสร้างรวมถึงจำนวน [90] และขนาดของชั้นซ่อน [91] การเรียนรู้ของ MLP จะใช้ข้อมูลการฝึกอบรมเพื่อปรับโครงข่ายให้ให้ผลลัพธ์ที่ถูกต้องขึ้นตามการวัดผล โครงสร้างของ MLP ประกอบด้วยสามส่วนหลักดังแสดงในภาพ 8



ภาพ 8 โครงสร้างของ Multilayer Perceptron

1) ชั้นนำเข้า (Input Layer) นี่คือชั้นแรกของ MLP ที่รับข้อมูลนำเข้า [92] เช่น คุณลักษณะเด่น (Features) ของข้อมูลที่ใช้ในการทำนายหรือประมวลผล แต่ละโหนดในชั้นนี้แทนคุณลักษณะเด่นอย่างหนึ่งและไม่มีการประมวลผลในชั้นนี้ ค่าที่ได้จากชั้นนำเข้าจะส่งต่อไปยังชั้นถัดไปโดยตรงโดยไม่มีการเปลี่ยนแปลง

2) ชั้นซ่อน (Hidden Layer) ชั้นนี้ประกอบด้วยหลายๆโหนดและมีการเชื่อมต่อกันในแต่ละโหนด [92, 93] แต่ละโหนดในชั้นซ่อนนี้มีความสามารถในการประมวลผลข้อมูล โดยแต่ละโหนดรับข้อมูลจากชั้นที่แล้วและส่งผลลัพธ์ต่อไปยังโหนดในชั้นถัดไป ชั้นซ่อนนี้เป็นส่วนที่ทำให้ MLP สามารถเรียนรู้โครงสร้างของข้อมูลและความสัมพันธ์ระหว่างคุณลักษณะได้ มีทั้งแบบชั้นซ่อนเดียว (Single Hidden Layer) และแบบหลายชั้น (Multiple Hidden Layers) ขึ้นอยู่กับโครงสร้างของโมเดล

3) ชั้นส่งออก [94] (Output Layer) ชั้นสุดท้ายนี้คือผลลัพธ์ที่ได้จากการประมวลผลข้อมูลในชั้นซ่อน และมีจำนวนโหนดตามจำนวนคลาสหรือค่าที่ต้องการทำนาย

MLP ทำงานโดยการเรียนรู้ค่าความสัมพันธ์ระหว่างข้อมูลนำเข้าและผลลัพธ์ โดยใช้การปรับค่าของน้ำหนักและการส่งผ่านข้อมูลผ่านชั้นต่างๆที่มีความซับซ้อนให้กลายเป็นผลลัพธ์ที่ต้องการโดยใช้ฟังก์ชันการเรียนรู้ เช่น Backpropagation [95] เพื่อปรับค่าของน้ำหนักเพื่อให้ผลลัพธ์ที่คำนวณได้อย่างถูกต้อง

2.4 งานวิจัยที่เกี่ยวข้อง

การระบาดของโควิด-19 ได้สร้างผลกระทบที่กว้างขวางต่อสุขภาพและเศรษฐกิจทั่วโลก ทำให้การคาดการณ์การแพร่กระจายของโรคเป็นสิ่งที่สำคัญอย่างยิ่ง ระบบสารสนเทศภูมิศาสตร์ และการเรียนรู้ของเครื่อง ได้รับความสนใจอย่างมากในฐานะเครื่องมือที่สามารถช่วยในการคาดการณ์และติดตามการแพร่กระจายของโควิด-19 การทบทวนวรรณกรรมนี้มุ่งเน้นการวิเคราะห์การใช้ GIS และการเรียนรู้ของเครื่องในการทำนายโควิด-19

GIS เป็นเครื่องมือที่มีประสิทธิภาพในการวิเคราะห์และแสดงข้อมูลทางภูมิศาสตร์ ซึ่งช่วยในการติดตามการแพร่กระจายของโควิด-19 ด้วยการรวบรวมข้อมูลจากแหล่งต่างๆ เช่น ข้อมูลการติดเชื้อรายวัน การเสียชีวิตรายวัน ความหนาแน่นของประชากรแต่ละจังหวัด GIS สามารถสร้างแผนที่และการวิเคราะห์ที่แสดงถึงพื้นที่เสี่ยงสูงและการเปลี่ยนแปลงของการแพร่กระจายได้ [96] งานวิจัยที่สำคัญเช่นการศึกษาโดย [97] แสดงให้เห็นว่า GIS สามารถใช้ในการสร้างแผนที่ความร้อนที่แสดงถึงการแพร่กระจายของโควิด-19 และช่วยในการวางแผนการตอบสนองของสาธารณสุข การใช้ GIS ยังมีการพัฒนาเพิ่มเติมในการวิเคราะห์เชิงลึก เช่น การใช้ข้อมูลที่เก็บรวบรวมจากโซเชียลมีเดียและแหล่งข้อมูลอื่นๆ เพื่อติดตามการเปลี่ยนแปลงของพฤติกรรมและการเคลื่อนไหวของประชากร ซึ่ง

สามารถให้ข้อมูลที่มีค่าในการคาดการณ์การแพร่กระจายในพื้นที่ต่างๆเพื่อการรวมข้อมูลนี้ช่วยเพิ่มความแม่นยำในการคาดการณ์และการตอบสนองที่เหมาะสม

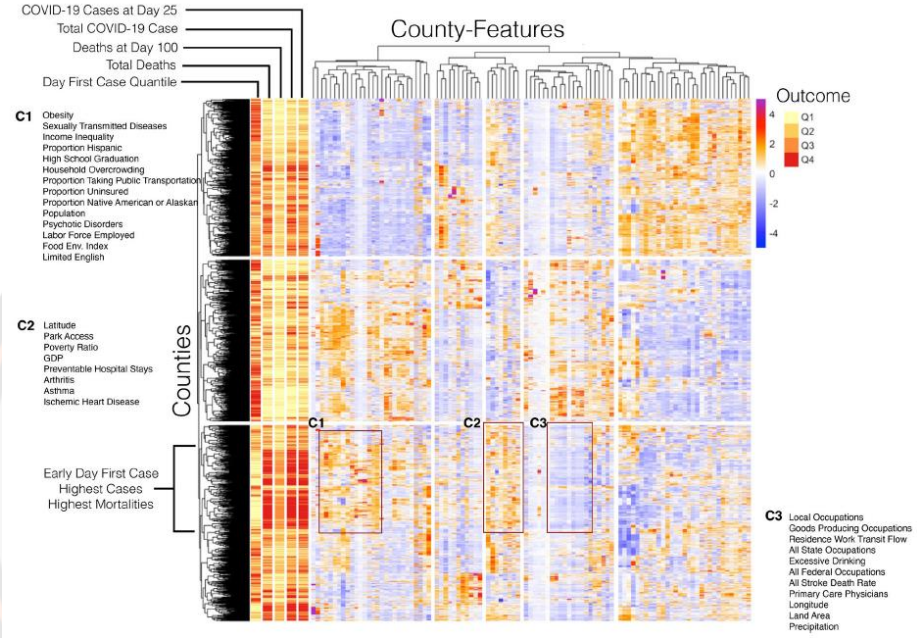
การเรียนรู้ของเครื่องได้แสดงให้เห็นถึงความสามารถในการวิเคราะห์และคาดการณ์ข้อมูลโควิด-19 อย่างมีประสิทธิภาพ เทคนิคต่างๆ เช่น การเรียนรู้เชิงลึกและอัลกอริธึมการถดถอย (Regression Algorithms) ใช้เพื่อสร้างโมเดลที่สามารถคาดการณ์การติดเชื้อในอนาคตได้ [98] งานวิจัย [99] ได้แสดงให้เห็นว่าโมเดลการเรียนรู้เชิงลึกสามารถคาดการณ์การติดเชื้อได้แม่นยำมากขึ้นโดยใช้ข้อมูลจากหลายแหล่ง เช่น ข้อมูลการเดินทางและข้อมูลประชากร การเรียนรู้เชิงลึกยังมีบทบาทสำคัญในด้านการประมวลผลภาพ เช่น การวิเคราะห์ภาพถ่ายทางการแพทย์เพื่อการวินิจฉัยโควิด-19 งานวิจัยโดย [100] พบว่าโมเดลการเรียนรู้เชิงลึกเช่น Convolutional Neural Networks (CNNs) สามารถใช้ในการตรวจจับและวิเคราะห์ภาพเอกซเรย์ของปอดเพื่อระบุลักษณะของโควิด-19 ได้อย่างแม่นยำ การศึกษาโดย [101] แสดงให้เห็นว่า CNNs สามารถแยกแยะระหว่างภาพ X-ray ของผู้ติดเชื้อโควิด-19 และผู้ป่วยที่มีโรคทางเดินหายใจอื่นๆ ได้ด้วยความแม่นยำที่สูง การใช้เทคนิคนี้ช่วยให้แพทย์สามารถวินิจฉัยโรคได้เร็วขึ้นและแม่นยำมากขึ้น

นอกจากนี้ เทคนิคการเรียนรู้เชิงลึกได้นำมาใช้ในการทำนายการแพร่กระจายของโควิด-19 โดยการวิเคราะห์ข้อมูลเชิงเวลา เช่น ข้อมูลการติดเชื้อรายวันและข้อมูลการเคลื่อนไหวของประชากร โมเดล RNN และ LSTM ได้รับการใช้ในการทำนายแนวโน้มการติดเชื้อในอนาคต การวิจัยโดย [102] แสดงให้เห็นว่า LSTM สามารถจับทิศทางการเปลี่ยนแปลงของการแพร่กระจายและคาดการณ์จำนวนผู้ติดเชื้อในอนาคตได้อย่างแม่นยำ โดยการวิเคราะห์ข้อมูลที่มีความเป็นระเบียบและไม่มีลำดับ

การรวมกันของ GIS และการเรียนรู้ของเครื่องสามารถเพิ่มความสามารถในการคาดการณ์โดยการใช้ข้อมูลเชิงพื้นที่และเชิงเวลาในการวิเคราะห์ การผสมผสานนี้ช่วยให้สามารถสร้างโมเดลที่มีความแม่นยำสูงขึ้นและสามารถตอบสนองได้ดีขึ้น [103] การศึกษาของ [104] พบว่า การรวม GIS กับการเรียนรู้ของเครื่องช่วยในการระบุจุดแผนภูมิความร้อนที่เกิดขึ้นใหม่และปรับปรุงการวางแผนการตอบสนองด้านสาธารณสุข โดยเฉพาะในการจัดสรรทรัพยากรและการกำหนดกลยุทธ์การควบคุม การรวมกันนี้ยังสามารถใช้ในการวิเคราะห์ผลกระทบของมาตรการควบคุมต่างๆ เช่น การปิดเมืองและการสวมหน้ากาก การวิจัยของ [105] แสดงให้เห็นว่า การรวมข้อมูล GIS กับการวิเคราะห์ ML ช่วยให้สามารถประเมินประสิทธิภาพของมาตรการเหล่านี้ได้ดีขึ้น และปรับกลยุทธ์ให้เหมาะสมตามสถานการณ์ที่เปลี่ยนแปลง

การใช้การเรียนรู้แบบรวมกันเพื่อศึกษาปัจจัยที่มีอิทธิพลต่อโควิด-19 ในแต่ละเขตของสหรัฐอเมริกา [106] ทำการวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 ซึ่งเป็นโรคติดต่อทางระบบทางเดินหายใจที่เกิดจากเชื้อไวรัส SARS-CoV-2 ซึ่งมีการแพร่กระจายอย่างรวดเร็วทั่วโลกตั้งแต่เดือนธันวาคม 2019 ในงานวิจัยนักวิจัยได้ศึกษาปัจจัยด้านสุขภาพ สังคม และสิ่งแวดล้อมที่มีผลกระทบต่อ

การแพร่ระบาดของโควิด-19 และอัตราการเสียชีวิตของสหรัฐอเมริกา โดยใช้ข้อมูลสุขภาพจิตและสุขภาพกาย มลพิษทางสิ่งแวดล้อม การเข้าถึงบริการสุขภาพ ลักษณะประชากร และข้อมูลทางระบาดวิทยา เพื่อสร้างชุดข้อมูลที่ใช้ในการวิเคราะห์ผลกระทบของโควิด-19 ในแต่ละเขต การใช้วิธีการเรียนรู้ของเครื่องแบบรวมช่วยในการระบุปัจจัยที่สำคัญที่สุดที่มีผลต่อการแพร่ระบาดและอัตราการเสียชีวิตจากโควิด-19 ผลการวิเคราะห์ข้อมูลแสดงในภาพ 9



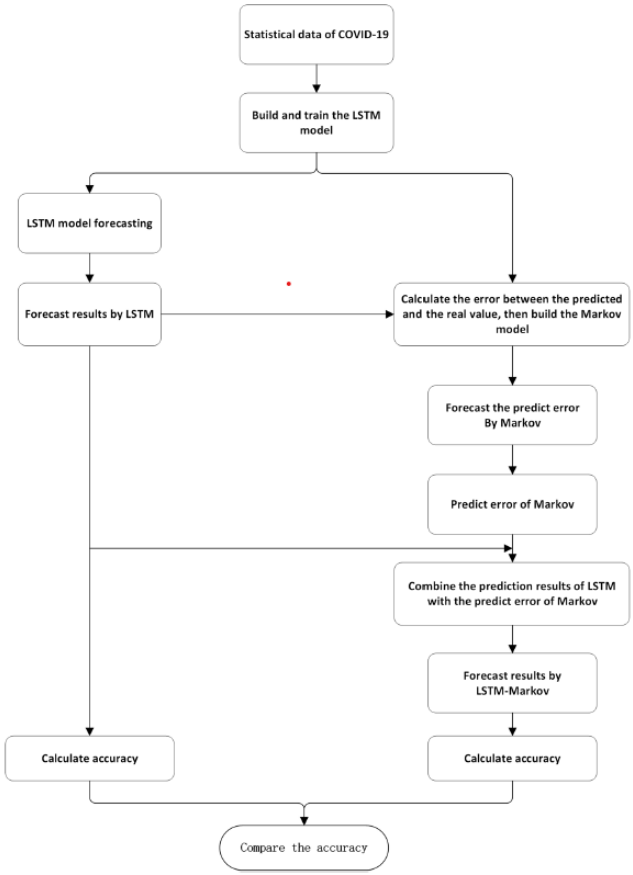
ภาพ 9 Heatmap การกระจายข้อมูลตามระดับเขตที่จัดกลุ่มโดย Dendrogram [106]

ภาพ 9 แสดงผลการวิเคราะห์ข้อมูลโควิด-19 ในระดับเขต (County) ของสหรัฐอเมริกาในรูปแบบแผนภูมิความร้อน ซึ่งจัดกลุ่มเขตต่าง ๆ ตามลักษณะเฉพาะของแต่ละเขต เช่น ปัจจัยทางสุขภาพ ประชากร และเศรษฐกิจ โดยใช้สีที่แตกต่างกันเพื่อแสดงค่าที่สูงหรือต่ำของแต่ละลักษณะ แผนภูมิความร้อนนี้ช่วยให้เห็นภาพรวมของความเชื่อมโยงระหว่างลักษณะต่าง ๆ กับผลลัพธ์ของการระบาดของโควิด-19 ในเขตต่าง ๆ จากแผนภูมิความร้อนนี้ พบว่าบางเขตที่มีการระบาดของโควิด-19 สูงมักมีปัจจัยร่วม เช่น การใช้ขนส่งสาธารณะ และสัดส่วนประชากรที่เป็นคนผิวสีหรือคนอเมริกันเชื้อสายแอฟริกาสูง ข้อมูลนี้ชี้ให้เห็นถึงบทบาทของปัจจัยทางสังคมและเศรษฐกิจในการกำหนดผลลัพธ์ของการระบาดในพื้นที่ต่าง ๆ ของประเทศ

งานวิจัย [106] ใช้วิธีการเรียนรู้ของเครื่องแบบ Super Learner ซึ่งประกอบด้วยอัลกอริทึมหลายตัวในการคาดการณ์ผลลัพธ์ของโควิด-19 ในแต่ละรัฐของสหรัฐอเมริกา ข้อมูลที่ใช้ในการ

วิเคราะห์ได้มาจากแหล่งข้อมูลต่างๆ เช่น CDC U.S. Census Bureau และ Google Mobility Data การวิเคราะห์ประกอบด้วยห้าผลลัพธ์ของโควิด-19 ได้แก่ วันที่มีผู้ป่วยรายแรกในเขตเมือง จำนวนผู้ป่วย 25 วันหลังจากพบผู้ป่วยรายแรก จำนวนผู้เสียชีวิตทั้งหมด 100 วันหลังจากพบผู้ป่วยรายแรก จำนวนผู้ป่วยทั้งหมด และจำนวนผู้เสียชีวิตทั้งหมดในเขตเมืองตั้งแต่เริ่มการระบาด ผลการวิจัยพบว่าตัวแปรที่สำคัญที่สุดในการทำนายจำนวนผู้ป่วยและผู้เสียชีวิตจากโควิด-19 ในเขตเมืองของสหรัฐอเมริกา ได้แก่ ประชากรกลุ่มชาติพันธุ์ที่เป็นชนกลุ่มน้อย การใช้ระบบขนส่งสาธารณะ และโรคที่สามารถป้องกันได้ การวิเคราะห์พบว่าประชากรกลุ่มชาติพันธุ์ที่เป็นชนกลุ่มน้อยและการใช้ระบบขนส่งสาธารณะสูงมีความเสี่ยงสูงในการแพร่ระบาดและการเสียชีวิตจากโควิด-19 การทำนายผลลัพธ์โดยการปรับตัวแปรเหล่านี้พบว่า การลดการใช้ระบบขนส่งสาธารณะลงร้อยละ 10 ช่วยลดจำนวนผู้เสียชีวิตได้ถึง 2,012 ราย และการเพิ่มประชากรกลุ่มชาติพันธุ์ที่เป็นชนกลุ่มน้อยในมณฑลร้อยละ 10 ทำให้จำนวนผู้เสียชีวิตเพิ่มขึ้นถึง 2,067 ราย การใช้วิธีการเรียนรู้ของเครื่องแบบ Super Learner ช่วยให้สามารถทำนายผลลัพธ์ได้อย่างแม่นยำ เนื่องจากใช้หลายอัลกอริทึมในการทำนายและปรับปรุงผลลัพธ์ให้ดีที่สุด การรวบรวมข้อมูลจากแหล่งต่างๆ ทำให้สามารถวิเคราะห์ปัจจัยที่มีผลต่อการแพร่ระบาดของโควิด-19 ได้อย่างครอบคลุมและละเอียด แต่ยังขาดปัจจัยอื่นๆในการคาดการณ์ที่สำคัญ เช่น การทำนายผู้ติดเชื้อรายวันหรือการเสียชีวิตรายวันที่เป็นส่วนสำคัญในการทำนายกับการแพร่ระบาดของเชื้อโควิด-19

การทำนายและวิเคราะห์แนวโน้มการระบาดของโควิด-19 โดยการรวมวิธี LSTM และ Markov [107] เป็นงานวิจัยที่ผสมผสานระหว่าง LSTM และ Markov method เพื่อเพิ่มความแม่นยำในการทำนายเนื่องจากการทำนายระยะกลางและระยะยาวโดยใช้ LSTM มักมีความคลาดเคลื่อนสูง โดยงานวิจัย [107] เสนอวิธีการผสมผสานระหว่าง LSTM และ Markov เพื่อแก้ไขปัญหานี้โดยใช้ข้อมูลจำนวนผู้ติดเชื้อโควิด-19 ที่ยืนยันแล้วจากสหรัฐอเมริกา อังกฤษ บราซิล และรัสเซีย โดยทำการฝึกโมเดล LSTM และสร้างเมทริกซ์การเปลี่ยนแปลงสถานะของ Markov เพื่อแก้ไขความคลาดเคลื่อนของการทำนาย ขั้นตอนการทำนายจำนวนผู้ติดเชื้อโควิด-19 แสดงในภาพ 10

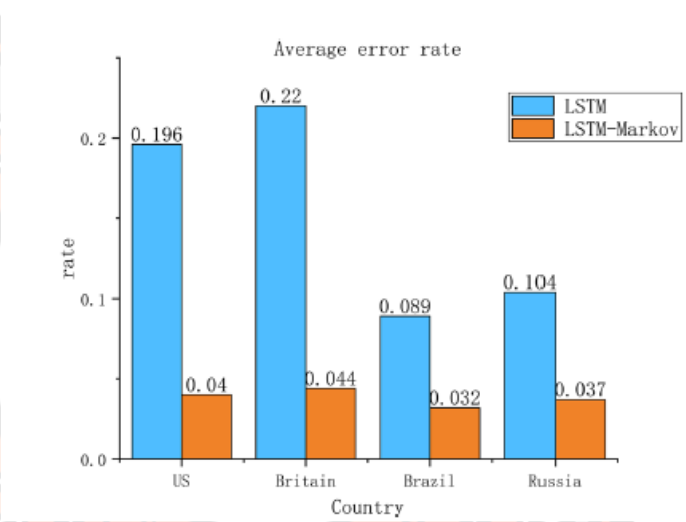


ภาพ 10 ขั้นตอนการคาดการณ์จำนวนผู้ติดเชื้อโควิด-19 โดยใช้ LSTM-Markov [107]

ภาพ 10 เป็นภาพสรุปขั้นตอนสำคัญในการคาดการณ์จำนวนผู้ป่วยโควิด-19 ที่ได้รับการยืนยัน โดยใช้โมเดลผสมผสานระหว่าง LSTM และ Markov กระบวนการนี้เริ่มต้นด้วยการใช้โมเดล LSTM ในการฝึกฝนกับข้อมูลจำนวนผู้ป่วยที่ได้รับการยืนยันจากสี่ประเทศหลัก ได้แก่ สหรัฐอเมริกา อังกฤษ บราซิล และรัสเซีย ซึ่งโมเดล LSTM นี้เป็นโมเดลการเรียนรู้เชิงลึกที่ได้รับการปรับปรุงมาจาก RNN และมีความสามารถในการจำข้อมูลที่มีลำดับเวลาได้ดี หลังจากที่ทำการฝึกโมเดล LSTM แล้ว ผลลัพธ์ที่ได้จากการคาดการณ์จำนวนผู้ป่วยที่ยืนยันแล้วจะนำไปเปรียบเทียบกับจำนวนผู้ป่วยจริงที่เกิดขึ้น จากนั้นความแตกต่างระหว่างจำนวนที่คาดการณ์ได้และจำนวนจริงจะนำมาคำนวณเพื่อสร้างเมริกซ์การถ่ายโอนความน่าจะเป็นในโมเดล Markov โมเดล Markov นี้เป็นโมเดลทางสถิติที่ใช้ในการคาดการณ์ผลลัพธ์ในอนาคต โดยอาศัยการเปลี่ยนแปลงสถานะของระบบในอดีต ขั้นตอนสุดท้ายใช้โมเดล LSTM เพื่อคาดการณ์จำนวนผู้ป่วยสะสมที่ได้รับการยืนยันในอนาคต จากนั้นใช้โมเดล Markov เพื่อปรับปรุงข้อผิดพลาดที่เกิดขึ้นในขั้นตอนการคาดการณ์ด้วย LSTM ผลลัพธ์สุดท้ายที่ได้

เป็นการพยากรณ์ที่ผ่านการแก้ไขความคลาดเคลื่อนทำให้การคาดการณ์มีความแม่นยำมากขึ้น โดยเฉพาะอย่างยิ่งในการพยากรณ์ระยะกลางและระยะยาว

การผสมผสานระหว่างโมเดล LSTM และ Markov ในการพยากรณ์นี้ช่วยลดข้อผิดพลาดที่เกิดขึ้นจากการคาดการณ์ด้วย LSTM เพียงอย่างเดียว ซึ่งเป็นการเพิ่มประสิทธิภาพและความแม่นยำในการคาดการณ์แนวโน้มการระบาดของโควิด-19 ในแต่ละประเทศ ผลการทดลองแสดงให้เห็นว่าโมเดล LSTM-Markov ที่ได้รับการปรับปรุงนี้มีประสิทธิภาพสูงกว่าโมเดล LSTM เพียงอย่างเดียวในการทำนายจำนวนผู้ป่วยในระยะเวลายาวนาน อัตราความผิดของโมเดล LSTM และโมเดล LSTM-Markov แสดงในภาพ 11

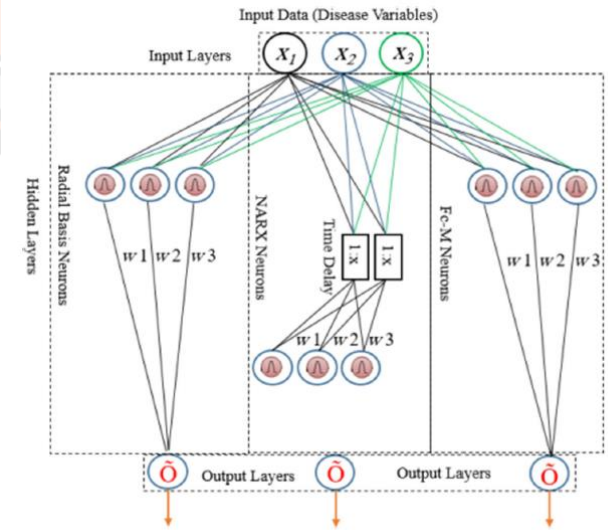


ภาพ 11 อัตราความผิดพลาดเฉลี่ยของโมเดล LSTM และโมเดล LSTM-Markov [107]

ภาพ 11 แสดงการเปรียบเทียบอัตราความผิดพลาดเฉลี่ยระหว่างโมเดล LSTM และ LSTM-Markov ในการพยากรณ์จำนวนผู้ป่วยโควิด-19 ที่ได้รับการยืนยันในสหรัฐอเมริกา อังกฤษ บราซิล และรัสเซีย ประกอบด้วยกราฟที่แสดงผลการพยากรณ์ของโมเดลทั้งสองและการเปรียบเทียบกับค่าจริงที่รายงานในสามช่วงเวลา ได้แก่ วันที่ 5 ธันวาคม 2563 5 มกราคม 2564 และ 5 กุมภาพันธ์ 2564 ผลการวิเคราะห์ที่แสดงในภาพ 11 ชี้ให้เห็นว่า โมเดล LSTM-Markov มีอัตราความผิดพลาดเฉลี่ยต่ำกว่าโมเดล LSTM อย่างมาก โดยในสหรัฐอเมริกา โมเดล LSTM มีอัตราความผิดพลาดเฉลี่ยประมาณ 0.152 ขณะที่โมเดล LSTM-Markov มีอัตราความผิดพลาดเฉลี่ยเพียง 0.038 ซึ่งต่ำกว่ามาก และสถานการณ์นี้เป็นจริงในทุกประเทศที่ทำการศึกษา

ภาพ 11 แสดงให้เห็นถึงประสิทธิภาพของโมเดล LSTM-Markov ที่ดีกว่าในการทำนายแนวโน้มการแพร่ระบาดของโควิด-19 โดยเฉพาะอย่างยิ่งเมื่อใช้ในการพยากรณ์ระยะกลางและระยะยาว ซึ่งอัตราความผิดพลาดที่ต่ำกว่านี้บ่งบอกว่าโมเดล LSTM-Markov มีความแม่นยำสูงกว่าและสามารถทำนายผลได้ใกล้เคียงกับความเป็นจริงมากกว่าการใช้โมเดล LSTM เพียงอย่างเดียว นอกจากนี้ โมเดลยังสามารถทำนายแนวโน้มการแพร่ระบาดได้ดียิ่งขึ้นในระยะกลางและระยะยาว การใช้ LSTM-Markov ช่วยลดความคลาดเคลื่อนของการทำนายและเพิ่มความแม่นยำในการทำนายแนวโน้มการแพร่ระบาดโดยการผสมผสาน Markov model ช่วยแก้ไขข้อจำกัดของ LSTM ในการทำนายระยะกลางและระยะยาวแต่งานวิจัย [107] ยังขาดการวิเคราะห์ข้อมูลในด้านต่างๆ เช่น การเสียชีวิตรายวัน การเสียชีวิตสะสม เนื่องจากการพัฒนาโมเดลนั้นมุ่งเน้นไปในส่วนของการทำนายจำนวนผู้ติดเชื้อรายวันเพียงอย่างเดียวเท่านั้นทำให้ยังขาดปัจจัยอื่นๆในการวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 ในด้านอื่นๆ

การวิเคราะห์การทำนายโควิด-19 โดยใช้กระบวนการปัญญาประดิษฐ์และการวิเคราะห์เชิงพื้นที่ GIS กรณีศึกษาในอิรัก [108] ได้ทำการทำนายการแพร่ระบาดของโควิด-19 ในอิรัก โดยใช้วิธีการทางปัญญาประดิษฐ์ (AI) และการวิเคราะห์พื้นที่ด้วยระบบภูมิสารสนเทศ (GIS) เพื่อช่วยในการวางแผนควบคุมการแพร่ระบาดให้มีประสิทธิภาพโดยการใช้เทคโนโลยีปัญญาประดิษฐ์ (AIT) และระบบภูมิสารสนเทศ (GIS) อาจช่วยในการวิเคราะห์และควบคุมการแพร่ระบาดของโรคนี้ได้ดีขึ้น งานวิจัย [107] ใช้โมเดลปัญญาประดิษฐ์สามแบบ



ภาพ 12 โครงสร้างของโมเดลที่นำเสนอใน RBF FCM และ NARX [108]

ภาพ 12 แสดงสถาปัตยกรรมของโมเดลที่พัฒนาเพื่อพยากรณ์การแพร่ระบาดของโควิด-19 ในอิรัก โดยโมเดลนี้ใช้การรวมกันของสามฟังก์ชันในโครงข่ายประสาทเทียม (Artificial Neural Networks - ANNs) ได้แก่ Radial Basis Function (RBF) Fuzzy Cluster Means (FCM) และ Non-linear Autoregressive Network with Exogenous Inputs (NARX) สถาปัตยกรรมของโมเดลนี้ออกแบบมาเพื่อให้สามารถคาดการณ์แนวโน้มการแพร่ระบาดได้อย่างแม่นยำ โดยพิจารณาจากข้อมูลการติดเชื้อ สถาปัตยกรรมของโมเดลประกอบด้วยขั้นตอนหลัก ๆ เช่น การเตรียมข้อมูล การเลือกฟังก์ชัน ANN ที่เหมาะสม การสร้างโมเดล การวิเคราะห์ทางสถิติ และการออกแบบส่วนต่อประสานผู้ใช้ การใช้ฟังก์ชัน RBF นั้นช่วยในการจัดการกับข้อมูลที่ไม่เป็นเชิงเส้น (Nonlinear Data) ส่วน FCM ใช้ในการจัดกลุ่มข้อมูลเพื่อระบุรูปแบบและแนวโน้มที่ซ่อนอยู่ ขณะที่ NARX ใช้ในการพยากรณ์ข้อมูลเชิงเวลา (Time-series) โดยผสมผสานข้อมูลภายนอกเข้ากับข้อมูลอดีตเพื่อนำมาประเมินผลในอนาคต ภาพ 12 ให้ภาพรวมของกระบวนการสร้างโมเดลและการเชื่อมโยงกันของฟังก์ชันต่าง ๆ ในการคาดการณ์การแพร่ระบาดของโควิด-19 โมเดลนี้นำไปทดสอบและประเมินประสิทธิภาพเพื่อให้แน่ใจว่าให้ผลลัพธ์ที่มีความแม่นยำและสามารถใช้งานได้จริงในสถานการณ์ที่ต้องพยากรณ์การแพร่ระบาดในอนาคต การเปรียบเทียบประสิทธิภาพของโมเดลแสดงในตาราง 1

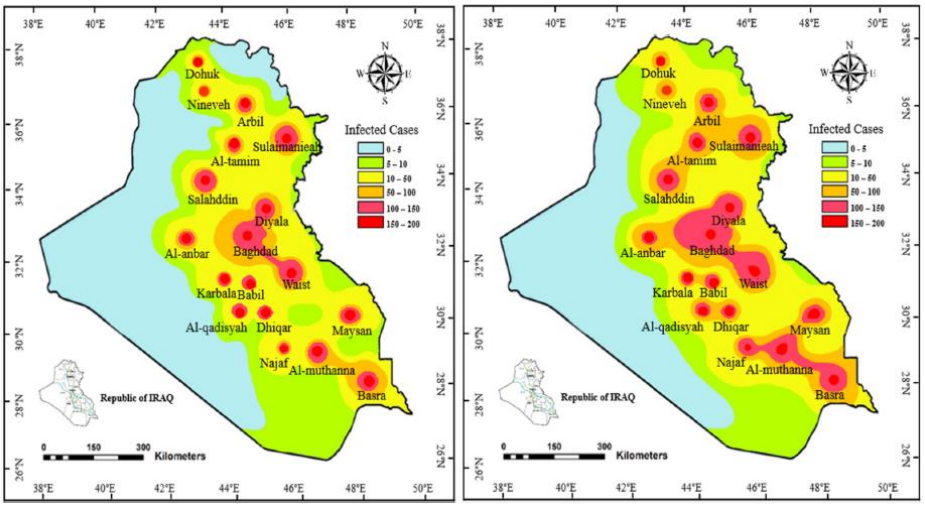
ตาราง 1 การเปรียบเทียบประสิทธิภาพของโมเดล [108]

Period	RBF	NARX	FCM	Predicted Average	Actual Dataset	% Error
June 1	6460	6451	6675	6528.6	6439	8.98
June 2	6853	6872	6899	6874.6	6868	7.84
June 3	7390	7377	7399	7388.6	7387	4.47
June 4	8182	8172	8177	8177	8168	3.20
June 5	8860	8850	8846	8852	8840	7.75
June 6	9855	9861	9875	9863.6	9846	13.50
June 7	11125	11012	11201	11112.6	11098	4.85
June 8	12345	12371	12382	12366	12366	3.10
June 9	13535	13512	13589	13545.3	13543	3.38
June 10	14295	14291	14281	14289.3	14268	2.25
June 11	15454	15442	15462	15452.6	15414	3.48
June 12	16715	16745	16736	16732	16675	3.42
June 13	17810	17799	17785	17798	17770	3.81
June 14	18982	18962	18967	18970.3	18950	7.56

ผลการเปรียบเทียบจำนวนผู้ติดเชื้อโควิด-19 ที่คาดการณ์ได้จากโมเดลที่พัฒนา [108] (ซึ่งประกอบด้วยฟังก์ชัน ANNs สามแบบ ได้แก่ RBF NARX และ FCM) กับข้อมูลจริง ผลลัพธ์ดังแสดง

ในตาราง 1 ใช้ในการวิเคราะห์ความถูกต้องของโมเดลโดยการคำนวณข้อผิดพลาดทางสถิติระหว่างจำนวนผู้ติดเชื้อที่คาดการณ์ได้กับจำนวนจริง

ข้อมูลในตาราง 1 แสดงให้เห็นถึงจำนวนผู้ติดเชื้อที่คาดการณ์ได้ในแต่ละวันตามโมเดล RBF NARX และ FCM พร้อมกับค่าเฉลี่ยของการคาดการณ์จากทั้งสามโมเดล และจำนวนผู้ติดเชื้อจริงที่รายงานในแต่ละวัน จากนั้นมีการคำนวณค่าร้อยละความผิดพลาดทางสถิติ ซึ่งค่าเฉลี่ยของร้อยละความผิดพลาดจากทั้งสามโมเดลในช่วงเวลานั้นอยู่ที่ประมาณร้อยละ 5.74 ซึ่งแสดงถึงความแม่นยำของโมเดลที่พัฒนา ข้อมูลนี้ช่วยให้เห็นว่าโมเดลที่พัฒนาขึ้น [108] มีความสามารถในการคาดการณ์จำนวนผู้ติดเชื้อได้อย่างแม่นยำ โดยมีความผิดพลาดเฉลี่ยที่ต่ำ แสดงถึงความน่าเชื่อถือของโมเดลในการใช้คาดการณ์การแพร่ระบาดของโควิด-19 ในอนาคต ผลการกระจายเชิงพื้นที่ตามจำนวนผู้ติดเชื้อจริงและผลการทำนายดังแสดงในภาพ 13



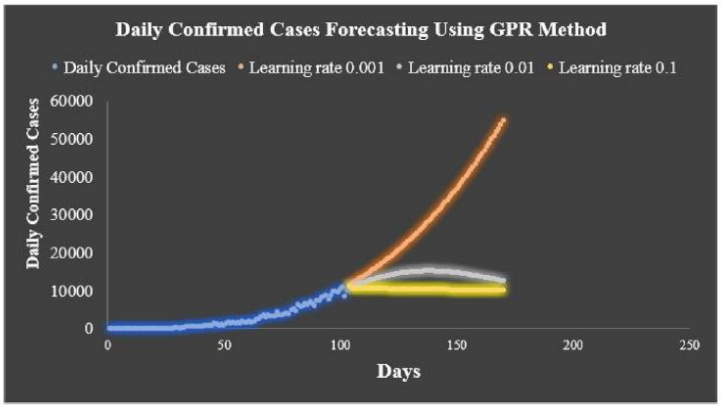
ภาพ 13 การกระจายเชิงพื้นที่ตามจำนวนผู้ติดเชื้อจริงและผลการทำนาย [108]

แผนที่การกระจายตัวเชิงพื้นที่ของผู้ติดเชื้อโควิด-19 ในประเทศอิรัก โดยแสดงการเปรียบเทียบระหว่างจำนวนผู้ติดเชื้อจริงและจำนวนผู้ติดเชื้อที่คาดการณ์ได้จากโมเดลที่พัฒนา แผนที่นี้สร้างขึ้นโดยใช้เครื่องมือ GIS เพื่อให้เห็นภาพการแพร่กระจายของโรคในพื้นที่ต่าง ๆ ทั่วประเทศอิรัก ภาพ 13 ประกอบด้วยแผนที่สองแผนที่ แผนที่แรกแสดงจำนวนผู้ติดเชื้อจริงที่บันทึกได้ในช่วงเวลาหนึ่ง ขณะที่แผนที่ที่สองแสดงจำนวนผู้ติดเชื้อที่คาดการณ์ได้ในช่วงเวลาเดียวกัน โดยใช้ข้อมูลจากโมเดลพยากรณ์โควิด-19 ที่พัฒนาขึ้น ผลการวิเคราะห์ที่แสดงในภาพ 13 ชี้ให้เห็นถึงการ

แพร่กระจายของโรคที่มีความเข้มข้นมากขึ้นในพื้นที่บางส่วนของอิรัก และแสดงให้เห็นว่ามีการเพิ่มขึ้นของผู้ติดเชื้อในระดับที่น่ากังวลในหลายพื้นที่

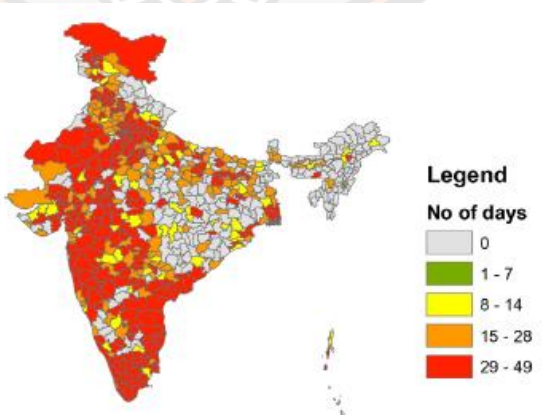
ผลการวิจัย [108] พบว่าโมเดลปัญญาประดิษฐ์ที่พัฒนาขึ้นมีประสิทธิภาพสูงในการคาดการณ์การแพร่ระบาดของโควิด-19 โดยมีประสิทธิภาพการทำงานสูงถึงร้อยละ 81.6 การใช้ GIS ในการวิเคราะห์และสร้างแผนที่การแพร่กระจายของเชื้อโรคช่วยให้สามารถประเมินความรุนแรงของการแพร่ระบาดในอิรักได้ในระยะสั้น (6 เดือน) ผลการคาดการณ์แสดงให้เห็นว่าความรุนแรงของการแพร่ระบาดจะเพิ่มขึ้นร้อยละ 17.1 และจำนวนผู้เสียชีวิตจะเพิ่มขึ้นร้อยละ 8.3 นอกจากนี้ โมเดลยังช่วยในการระบุพื้นที่ที่มีความเสี่ยงสูงและวางแผนการจัดการโรคได้อย่างมีประสิทธิภาพ การใช้ GIS ช่วยให้สามารถสร้างแผนที่การแพร่กระจายของเชื้อโรคและระบุพื้นที่ที่มีความเสี่ยงสูงได้อย่างมีประสิทธิภาพ การใช้เทคโนโลยีปัญญาประดิษฐ์และ GIS ร่วมกันเป็นเครื่องมือที่มีประสิทธิภาพในการคาดการณ์และจัดการการแพร่ระบาดของโควิด-19 อย่างไรก็ตาม การปรับปรุงและอัปเดตข้อมูลอย่างต่อเนื่องเป็นสิ่งสำคัญเพื่อลดความคลาดเคลื่อนในการทำนายการใช้ GIS ช่วยให้สามารถสร้างแผนที่การแพร่กระจายของเชื้อโรคและระบุพื้นที่ที่มีความเสี่ยงสูงได้อย่างชัดเจนผ่านการแสดงผลของการทำนายในส่วนของแผนภูมิความร้อนทำให้ผู้ใช้สามารถวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 ได้อย่างชัดเจนแต่อย่างไรก็ตามงานวิจัย [108] ยังขาดความหลากหลายของข้อมูลที่นำมาใช้ในการพัฒนาโมเดลสำหรับการทำนายซึ่งอาจจะทำให้เกิดการข้อผิดพลาดในการทำนายในปัจจัยต่างๆที่เป็นผลทำให้การทำนายนั้นมีความแม่นยำมากขึ้น

การคาดการณ์ความรุนแรงของการแพร่ระบาดของโควิด-19 ในอินเดียโดยใช้ GIS และวิธีการเรียนรู้ของเครื่อง [109] การทำนายการแพร่ระบาดของโควิด-19 ในอินเดีย โดยใช้วิธีการทางภูมิศาสตร์สารสนเทศ และการเรียนรู้ของเครื่อง งานวิจัย [109] วางแผนควบคุมการแพร่ระบาดให้มีประสิทธิภาพงานโดยใช้การเรียนรู้ของเครื่องสามแบบ ได้แก่ การใช้ Decision Tree การใช้ Support Vector Machine (SVM) และ การใช้ Gaussian Process Regression (GPR) ในการคาดการณ์จำนวนผู้ติดเชื้อโควิด-19 ในอนาคต เพื่อสร้างโมเดลการคาดการณ์จุดหักเห (Inflection Points) ซึ่งช่วยในการวางแผนเปิดล็อกดาวน์อย่างมียุทธศาสตร์ และใช้ดัชนีความรุนแรง (Criticality Index) ที่พัฒนาขึ้นเพื่อประเมินความรุนแรงในแต่ละพื้นที่ การทำนายจำนวนผู้หายป่วยสะสมโดยใช้วิธี Exponential GPR แสดงในภาพ 14



ภาพ 14 การทำนายจำนวนผู้หายป่วยสะสมโดยใช้วิธี Exponential GPR [109]

ผลการพยากรณ์จำนวนผู้ป่วยสะสมที่ได้รับการยืนยันของโควิด-19 โดยใช้วิธี GPR ซึ่งเป็นหนึ่งในเทคนิคการเรียนรู้ของเครื่อง ภาพ 14 สรุปผลการพยากรณ์ที่ครอบคลุมช่วงเวลาหลังวันที่ 10 มิถุนายน 2563 ไปอีกสองเดือน โดยคำนวณจากสถานการณ์ที่เลวร้ายที่สุดที่อาจเกิดขึ้น ภายใต้สมมติฐานที่ว่า การล็อกดาวน์ยังคงมีผลบังคับใช้อย่างเข้มงวด ผลการพยากรณ์จากวิธี Exponential GPR นี้แสดงให้เห็นว่า จำนวนผู้ป่วยสะสมมีแนวโน้มที่เพิ่มขึ้นอย่างต่อเนื่อง จนกระทั่งถึงช่วงปลายเดือนกรกฎาคม 2563 ซึ่งคาดว่าอัตราการเพิ่มขึ้นของผู้ป่วยใหม่ต่อวันเริ่มลดลง หากมาตรการล็อกดาวน์ยังคงมีประสิทธิภาพในการควบคุมการแพร่ระบาด ภาพ 14 ช่วยให้เห็นถึงความรุนแรงของการระบาดที่อาจเกิดขึ้น หากไม่มีการดำเนินมาตรการเพิ่มเติมหรือการปรับปรุงมาตรการป้องกันต่าง ๆ จุดเปลี่ยนแปลงจำนวนวันของแต่ละเขตเริ่มจาก 2 สิงหาคม แสดงในภาพ 15



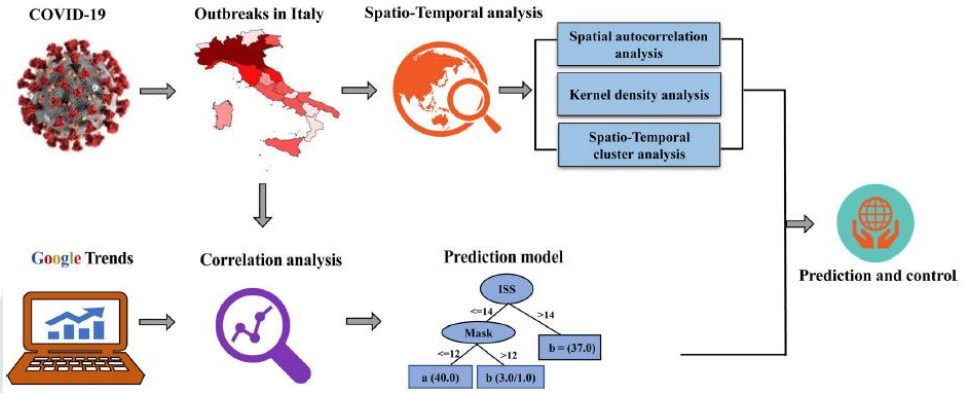
ภาพ 15 จุดเปลี่ยนแปลงจำนวนวันของแต่ละเขตเริ่มจาก 2 สิงหาคม [109]

การคาดการณ์จุดหักเหของการระบาดของโควิด-19 ในแต่ละเขตของประเทศอินเดีย พบจุดหักเห (Inflection Point) ซึ่งหมายถึงจุดที่อัตราการเพิ่มขึ้นของผู้ป่วยรายใหม่ต่อวันเริ่มลดลง เป็นสัญญาณบ่งชี้ว่าแนวโน้มการระบาดอาจเริ่มชะลอตัวลง ภาพ 15 จัดทำขึ้นเพื่อวิเคราะห์ช่วงเวลาที่คาดว่าการระบาดจะถึงจุดหักเหในแต่ละเขต โดยนับจากวันที่ 2 สิงหาคม 2563 ในการวิเคราะห์นี้ เขตต่าง ๆ ได้จัดกลุ่มตามจำนวนวันหลังจากวันที่ 2 สิงหาคม ที่คาดว่าจุดหักเหจะเกิดขึ้น เขตที่มีจุดหักเหเร็วกว่าจะอยู่ในกลุ่มแรก และสามารถพิจารณาปลดล็อกดาวน์หรือผ่อนคลายมาตรการควบคุมได้ก่อน ขณะที่เขตที่คาดว่าจะถึงจุดหักเหช้ากว่า จะต้องรักษามาตรการควบคุมการระบาดต่อไป ภาพ 15 ช่วยให้ผู้กำหนดนโยบายสามารถมองเห็นภาพรวมของการระบาดในระดับเขต และสามารถวางแผนการเปิดเมืองหรือเขตพื้นที่ต่าง ๆ ได้อย่างเป็นลำดับขั้นตอนตามการคาดการณ์ที่แสดงในภาพ 15 และยังบ่งบอกถึงความแตกต่างในการระบาดของโควิด-19 ระหว่างเขตต่าง ๆ ในอินเดีย โดยบางเขตคาดว่าจะถึงจุดหักเหเร็วกว่าเขตอื่น ๆ ซึ่งสะท้อนถึงประสิทธิภาพของมาตรการควบคุมที่แตกต่างกันไปในแต่ละพื้นที่ นอกจากนี้ การวิเคราะห์จุดหักเหนี้ยังช่วยให้สามารถคาดการณ์ได้ว่าควรเน้นการบังคับใช้มาตรการเข้มงวดเพิ่มเติมในเขตใด และสามารถผ่อนคลายมาตรการในเขตใดได้บ้าง เพื่อลดผลกระทบทางเศรษฐกิจและสังคมที่เกิดจากการล็อกดาวน์

โดยดัชนีนี้คำนึงถึงจำนวนผู้ติดเชื้อ การเตรียมความพร้อมของระบบสุขภาพ และความหนาแน่นของประชากร ผลการทดลองพบว่า GPR มีประสิทธิภาพสูงสุดในการคาดการณ์การแพร่ระบาดของโควิด-19 ในอินเดีย โดยมีค่า R-squared สูงถึง 0.95 ซึ่งสูงกว่า SVM และ Decision Tree ที่มีค่า R-squared เท่ากับ 0.93 และ 0.94 ตามลำดับ นอกจากนี้ ดัชนีความรุนแรงที่พัฒนาขึ้นช่วยให้สามารถจำแนกพื้นที่ตามระดับความเสี่ยงได้อย่างมีประสิทธิภาพ ทำให้สามารถวางแผนการเปิดล็อกดาวน์และการจัดการทรัพยากรได้อย่างเหมาะสม สรุปได้ว่าการใช้ GIS และการเรียนรู้ของเครื่องเป็นเครื่องมือที่มีประสิทธิภาพในการคาดการณ์และจัดการการแพร่ระบาดของโควิด-19 การใช้ GPR มีความแม่นยำในการคาดการณ์สูง ทำให้การวางแผนและการจัดการทรัพยากรมีประสิทธิภาพ สามารถระบุจุดหักเหของการระบาดได้อย่างแม่นยำเนื่องจากสามารถระบุดัชนีความรุนแรงช่วยในการจำแนกพื้นที่ตามระดับความเสี่ยง ทำให้สามารถวางแผนการเปิดล็อกดาวน์ได้อย่างมียุทธศาสตร์ อย่างไรก็ตามงานวิจัย [109] ยังขาดความหลากหลายของข้อมูลที่เป็นปัจจัยสำคัญที่ใช้ในการทำนายอีกหลายส่วนเช่น ข้อมูลการฉีดวัคซีน ข้อมูลความหนาแน่นของประชากรในพื้นที่ เนื่องจากข้อมูลเหล่านี้มีผลสำคัญมากในการกำหนดหหรือวิเคราะห์การล็อกดาวน์

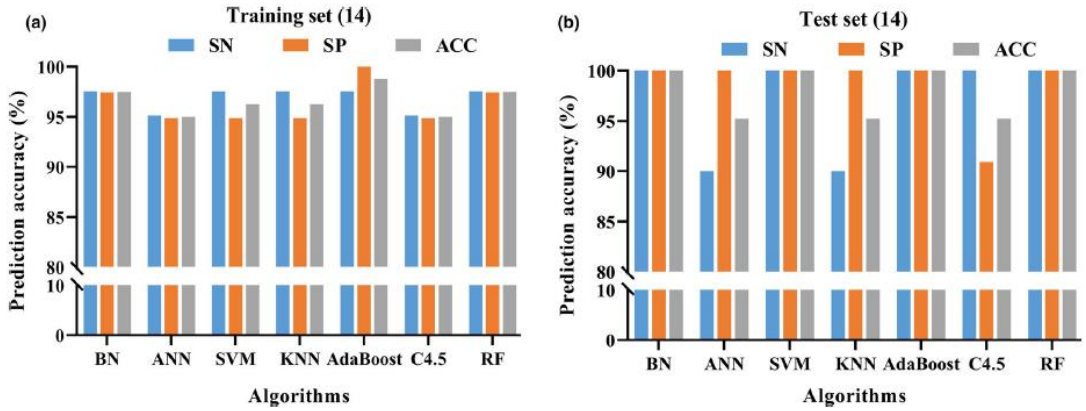
การวิเคราะห์การระบาดของโควิด-19 ในอิตาลีโดยอิงจากข้อมูลภูมิศาสตร์เชิงพื้นที่และเชิงเวลา [110] การวิเคราะห์การแพร่ระบาดของโควิด-19 ในอิตาลี โดยใช้ข้อมูลทางภูมิศาสตร์เชิงพื้นที่และข้อมูลจาก Google Trends เพื่อช่วยในการทำนายและจัดการการแพร่ระบาด [110] มุ่งเน้นการวิเคราะห์การแพร่กระจายของโรคในอิตาลีและการใช้ข้อมูลจาก Google Trends เพื่อช่วยในการ

ทำนายและจัดการการแพร่ระบาด [110] ใช้ข้อมูลจาก European Centre for Disease Prevention and Control (ECDC) และ Google Trends ข้อมูลจาก ECDC รวมถึงจำนวนผู้ติดเชื้อรายใหม่และผู้เสียชีวิตในแต่ละวันตั้งแต่วันที่ 31 ธันวาคม 2562 ถึง 30 กันยายน 2563 ข้อมูลจาก Google Trends มาจากการค้นหา 40 คำหลักในภูมิภาคอิตาลีตั้งแต่วันที่ 1 มกราคม 2563 ถึง 10 เมษายน 2563 และนำข้อมูลเหล่านี้มาวิเคราะห์ด้วยวิธีการทางภูมิสารสนเทศเชิงพื้นที่และการเรียนรู้ของเครื่อง ขั้นตอนการทำงานของงานโมเดลแสดงในภาพ 16



ภาพ 16 ขั้นตอนการทำงานของงานโมเดลที่ใช้ทำนาย [110]

การทำงานของวิธีการวิเคราะห์และคาดการณ์การระบาดของโควิด-19 ในประเทศอิตาลีใช้การผสมผสานระหว่างระบบสารสนเทศทางภูมิศาสตร์และอัลกอริธึมการเรียนรู้ของเครื่อง ภาพ 16 สรุปขั้นตอนต่าง ๆ ในการเก็บรวบรวมข้อมูล การวิเคราะห์ข้อมูลเชิงพื้นที่และเชิงเวลา และการสร้างโมเดลพยากรณ์โดยใช้ข้อมูลจาก Google Trends รวมถึงข้อมูลการระบาดจริง ในขั้นตอนแรกเก็บรวบรวมข้อมูลการระบาดของโควิด-19 จากแหล่งต่าง ๆ รวมถึงข้อมูลจาก Google Trends ได้นำมาใช้เพื่อทำการวิเคราะห์เชิงพื้นที่ โดยใช้เครื่องมือ GIS เพื่อสร้างแผนที่ความหนาแน่น (Kernel Density) และทำการวิเคราะห์การจัดกลุ่มทางภูมิศาสตร์ (Spatial Clustering) ขั้นตอนถัดไปคือการใช้ข้อมูลจาก Google Trends มาประมวลผลร่วมกับอัลกอริธึมการเรียนรู้ของเครื่อง เช่น อัลกอริธึม Adaboost อัลกอริธึม Random Forest และ อัลกอริธึม Bayesian Network เพื่อสร้างโมเดลพยากรณ์การระบาด โดยมีการคัดเลือกคุณลักษณะสำคัญที่ส่งผลต่อการแพร่ระบาดของโรคและแสดงภาพรวมของการเชื่อมโยงข้อมูลและการวิเคราะห์ที่ใช้ในการคาดการณ์การระบาดของโควิด-19 ในอิตาลี โดยใช้ทั้งข้อมูลทางภูมิศาสตร์และข้อมูลเชิงเวลาเพื่อสร้างความแม่นยำในการพยากรณ์และการวางแผนป้องกันการแพร่ระบาดในอนาคต ผลลัพธ์การพยากรณ์แสดงในภาพ 17



ภาพ 17 ผลลัพธ์การจำลองของอัลกอริธึมต่าง ๆ โดยใช้ 14 คำจากข้อมูล [110]

การจำลองการแพร่ระบาดของโควิด-19 ในประเทศอิตาลีใช้ข้อมูลจาก Google Trends และอัลกอริธึมต่าง ๆ เพื่อสร้างโมเดลที่คาดการณ์การระบาดของโรค ภาพ 17 นำเสนอผลลัพธ์การทดสอบความแม่นยำของอัลกอริธึมต่าง ๆ ที่ใช้ในการสร้างโมเดล ซึ่งประกอบไปด้วย อัลกอริธึม Adaboost อัลกอริธึม Random Forest อัลกอริธึม Bayesian Network อัลกอริธึม C4.5 อัลกอริธึม SVM และ อัลกอริธึม KNN ภาพ 17 แสดงให้เห็นว่าการทำนายผลการระบาดโดยใช้อัลกอริธึม Adaboost มีความแม่นยำสูงสุด โดยมีความแม่นยำในการทดสอบ ที่ร้อยละ 98.75 สำหรับชุดข้อมูลฝึกฝน และร้อยละร้อยสำหรับชุดข้อมูลทดสอบ เมื่อเทียบกับอัลกอริธึมอื่น ๆ ที่มีความแม่นยำต่ำกว่า นอกจากนี้ ค่า MCC (Matthews Correlation Coefficient) ซึ่งเป็นตัวชี้วัดความสัมพันธ์ระหว่างค่าที่ทำนายได้กับค่าจริงยังสูงสุดเมื่อใช้ Adaboost ผลการพยากรณ์แสดงในตาราง 2

ตาราง 2 ผลของจำลองด้วยฟีเจอร์เดียวโดยใช้ Adaboost algorithms [110]

Keywords	SN	SP	ACC	AUC
Mask	95.1	100.0	97.5	0.97
Pneumonia	97.6	97.4	97.5	0.96
Thermometer	97.6	92.3	95.0	0.95
ISS	97.6	92.3	95.0	0.95
Disinfect	97.6	92.3	95.0	0.94
Disposable gloves	95.1	76.9	86.3	0.90
Isolation	85.4	89.7	87.5	0.89
Epidemic	75.6	89.7	82.5	0.89
WHO	70.7	84.6	77.5	0.85
Fever	97.6	71.8	85.0	0.84

N95	80.5	69.2	75.0	0.80
Goggle	73.2	43.6	58.8	0.63
Medical supplies	95.1	15.4	63.8	0.59
Fatigue	56.1	71.8	63.8	0.57

ตาราง 2 นำเสนอผลลัพธ์การพยากรณ์โดยอิงจากคำสำคัญ (Keywords) ต่าง ๆ ที่เกี่ยวข้องกับการระบาดของโควิด-19 เช่น หน้ากากอนามัย (Mask) ปอดบวม (Pneumonia) เครื่องวัดอุณหภูมิ (Thermometer) การฆ่าเชื้อ (Disinfect) และถุงมือแบบใช้แล้วทิ้ง (Disposable Gloves) ตาราง 2 แสดงค่าตัวชี้วัดที่สำคัญ ได้แก่ ความไว (Sensitivity - SN) ความจำเพาะ (Specificity - SP) ความแม่นยำ (Accuracy - ACC) และพื้นที่ใต้กราฟ ROC (Area Under Curve - AUC) สำหรับคำสำคัญแต่ละคำ โดยผลลัพธ์แสดงว่า คำสำคัญ "Mask" มีค่า AUC สูงสุดที่ 0.97 ซึ่งบ่งบอกถึงความสามารถในการพยากรณ์ที่ดีมากของโมเดลเมื่อใช้คำสำคัญนี้ นอกจากนี้ คำว่า "Pneumonia" และ "Thermometer" ก็มีค่า AUC ที่สูงเช่นกัน ที่ 0.96 และ 0.95 ตามลำดับ จากผลลัพธ์ที่แสดงในตาราง 2 สรุปได้ว่า คำสำคัญเหล่านี้มีบทบาทสำคัญในการสร้างโมเดลพยากรณ์การระบาดของโควิด-19 และสามารถช่วยเพิ่มความแม่นยำในการพยากรณ์เมื่อใช้ร่วมกับ Adaboost algorithm

ผลการวิจัยของ [110] พบว่า การแพร่ระบาดของโควิด-19 ในอิตาลีมีแนวโน้มทางเวลาและการกระจายตัวทางภูมิศาสตร์ที่ชัดเจน โดยการแพร่ระบาดมุ่งเน้นไปที่ภาคเหนือของอิตาลีและค่อยๆ แพร่กระจายไปยังภูมิภาคอื่นๆ นอกจากนี้ การใช้ Google Trends ในการทำนายการแพร่ระบาดแสดงให้เห็นว่าคำค้นหาเกี่ยวกับการสวมใส่หน้ากาก การฆ่าเชื้อ และถุงมือยาง มีความสัมพันธ์สูงกับจำนวนผู้ติดเชื้อรายวัน โดยค่า AUC ของการทำนายโดยใช้ Adaboost algorithm มีค่าสูงกว่า 0.9 ซึ่งบ่งชี้ว่าคุณสมบัติเหล่านี้มีส่วนช่วยอย่างมากในการสร้างโมเดลการทำนาย การใช้ Adaboost algorithm ในการทำนายมีความแม่นยำสูง ทำให้สามารถทำนายแนวโน้มการแพร่ระบาดได้อย่างแม่นยำในระดับหนึ่งและการใช้ข้อมูลการค้นหาจาก Google Trends ช่วยให้เห็นภาพรวมของความสนใจและการรับรู้ของประชาชนต่อการแพร่ระบาด ในการพัฒนาโมเดลของงานวิจัย [110] มีการใช้ข้อมูลที่หลากหลายและโมเดลที่พัฒนานั้นยังมีความแม่นยำในการทำนายเช่นกัน ในส่วนของการแสดงผลของแผนที่นั้นมีการทำแผนภูมิความร้อนมาใช้ในการแสดงผลทำให้การแสดงผลมีความละเอียดและแสดงผลได้อย่างชัดเจน

จากการทบทวนงานวิจัยที่เกี่ยวข้องผู้วิจัยได้เลือก 5 งานวิจัย เพื่อเปรียบเทียบประสิทธิภาพของโมเดลที่ออกกแบบและพัฒนาขึ้นในงานวิจัยนี้ ตาราง 3 เปรียบเทียบงานวิจัยทั้ง 5 คือ [106-110] โดยแต่ละงานวิจัยนั้นมีจุดเด่นในหลายด้านที่แตกต่างกันออกไป จากการวิเคราะห์งานวิจัยทั้ง 5 พบว่ามีการใช้การเรียนรู้ของเครื่องหรือการเรียนรู้เชิงลึกร่วมกันกับข้อมูลทางด้านสารสนเทศภูมิศาสตร์ที่

สามารถทำให้การทำนายหรือการวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 แม่นยำมากขึ้น งานวิจัยนี้มุ่งเน้นไปในส่วนของการทำนายการแพร่ระบาดของเชื้อโควิด-19 โดยใช้ข้อมูลของประเทศไทยในการพัฒนาและปรับปรุงโมเดล (Fine-Tune) ให้มีความแม่นยำรวมไปถึงการสร้างแผนที่เพื่อรองรับข้อมูลที่ได้จากผลการทำนายโดยอยู่ในรูปแบบของเว็บแอปพลิเคชัน

ตาราง 3 สรุป 5 งานวิจัยที่นำมาเปรียบเทียบ

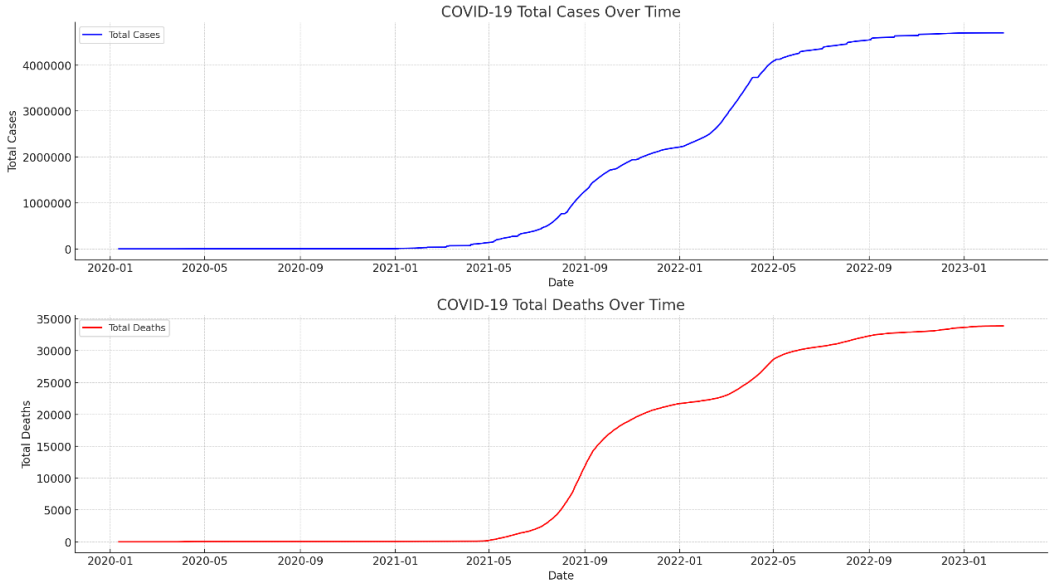
งานวิจัย	แหล่งที่มา	โมเดล	ความแม่นยำ	GIS	ปี
[110]	Wiley Online Library	• Baysian • Adaboost • SVM • KNN • Random forest • C4.5 • ANN	95.24%	✓	2021
[107]	Nature	• LSTM+Markov	96%	✓	2021
[108]	Springer	• RBF • FCM • NARX	81.6%	✓	2020
[109]	Science Direct	• GPR • SVM • Decision tree	95%	✓	2021
[106]	Nature	• Super Learner Ensemble	87%	✓	2021

บทที่ 3
วิธีการดำเนินงานวิจัย

สถาปัตยกรรมของระบบทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 นี้แบ่งออกเป็นสามโมดูลหลักที่มีหน้าที่แตกต่างกันเพื่อประสิทธิภาพสูงสุดในการทำงาน โมดูลที่ 1 ทำหน้าที่จัดการข้อมูล โดยเริ่มจากการเก็บรวบรวมข้อมูลจากฐานข้อมูลของกรมควบคุมโรค (MoPH) และดำเนินการจัดเตรียมข้อมูลก่อนเข้าสู่กระบวนการประมวลผล เช่น การทำความสะอาดข้อมูลและการปรับปรุงคุณภาพของข้อมูล โมดูลที่ 2 เป็นกระบวนการสร้างโมเดลทำนาย โดยใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งประกอบด้วย LSTM และ MLP ที่ทำงานร่วมกันเพื่อทำนายการแพร่ระบาดของเชื้อโควิด-19 ซึ่งผลลัพธ์การทำนายจะส่งต่อไปยังโมดูลที่ 3 ที่ทำหน้าที่แสดงผลการทำนาย โดยใช้ระบบ Geographic Information System (GIS) เพื่อสร้างแผนที่แสดงผลที่ชัดเจนและเข้าใจง่าย

3.1 ข้อมูลสถานการณ์โควิดในประเทศไทย

การสำรวจข้อมูลจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสมจากโควิด-19 ของทุกจังหวัดในประเทศไทยแบบรวมกัน ตั้งแต่เดือนมกราคม พ.ศ. 2563 ถึงมกราคม พ.ศ. 2566 เป็นกระบวนการที่สำคัญในการทำความเข้าใจแนวโน้มการระบาดของโรคและผลกระทบต่อประชากร ข้อมูลที่รวบรวมได้ประกอบด้วยจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสม ดังภาพ 18 ซึ่งนำเสนอในรูปแบบกราฟเพื่อแสดงการเปลี่ยนแปลงตามเวลา ข้อมูลที่ได้มานั้นจะทำการจัดระเบียบในรูปแบบที่สามารถนำไปใช้ในการวิเคราะห์เพิ่มเติมได้ การมีข้อมูลที่ครบถ้วนและถูกต้องช่วยให้สามารถประเมินสถานการณ์และตัดสินใจในด้านการจัดการและการป้องกันการระบาดในอนาคตได้อย่างมีประสิทธิภาพ



ภาพ 18 แนวโน้มของการติดเชื้อสะสมและการเสียชีวิตสะสม

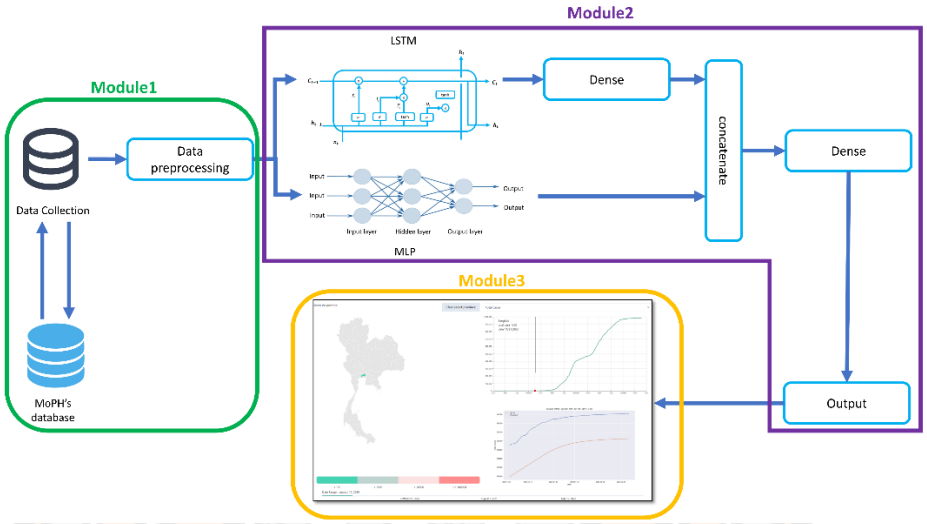
ภาพ 18 กราฟเส้นสีนำเงินแสดงจำนวนผู้ติดเชื้อโควิด-19 สะสมในประเทศตั้งแต่เดือนมกราคม พ.ศ. 2563 ถึงมกราคม พ.ศ. 2566 โดยใช้เส้นสีน้ำเงินแทนข้อมูลจำนวนผู้ติดเชื้อทั้งหมด กราฟแสดงการเพิ่มขึ้นของจำนวนผู้ติดเชื้ออย่างชัดเจน โดยเริ่มต้นที่จำนวนผู้ติดเชื้อน้อยมากในช่วงต้นปี พ.ศ. 2563 และค่อยๆ เพิ่มขึ้นจนถึงกลางปี พ.ศ. 2564 หลังจากนั้นมีการเพิ่มขึ้นอย่างรวดเร็วและต่อเนื่องในปี พ.ศ. 2564 และต้นปี พ.ศ. 2565 ก่อนที่จะชะลอตัวและคงที่ในช่วงปลายปี พ.ศ. 2565 และต้นปี พ.ศ. 2566 เส้นกราฟนี้มีลักษณะโค้งขึ้นอย่างชันในช่วงที่มีการระบาดหนักและแสดงถึงการสะสมของผู้ติดเชื้อที่เพิ่มขึ้นตามเวลาเนื่องจากเป็นการติดเชื้อสะสม

ภาพ 18 กราฟเส้นสีแดงแสดงจำนวนผู้เสียชีวิตจากโควิด-19 สะสมในประเทศในช่วงเวลาเดียวกัน โดยใช้เส้นสีแดงแทนข้อมูลจำนวนผู้เสียชีวิตทั้งหมด กราฟนี้แสดงให้เห็นการเพิ่มขึ้นของจำนวนผู้เสียชีวิตอย่างค่อยเป็นค่อยไปในช่วงต้นปี พ.ศ. 2563 และคงที่อยู่ระยะหนึ่งก่อนที่จะเพิ่มขึ้นอย่างชัดเจนในช่วงกลางปี พ.ศ. 2564 ถึงกลางปี พ.ศ. 2565 เส้นกราฟนี้มีลักษณะโค้งขึ้นอย่างนุ่มนวลกว่าเมื่อเทียบกับกราฟจำนวนผู้ติดเชื้อ ซึ่งแสดงให้เห็นว่าจำนวนผู้เสียชีวิตเพิ่มขึ้นอย่างต่อเนื่องแต่ไม่เร็วเท่ากับจำนวนผู้ติดเชื้อ หลังจากช่วงกลางปี พ.ศ. 2565 จำนวนผู้เสียชีวิตเริ่มคงที่และมีการเพิ่มขึ้นน้อยมากในช่วงปลายปี พ.ศ. 2565 และต้นปี พ.ศ. 2566

งานวิจัยนี้พัฒนาโมเดลเพื่อทำนายจำนวนผู้ติดเชื้อโควิด-19 และผู้เสียชีวิตจากโควิด-19 โดยทำนายข้อมูลรายจังหวัด และพัฒนาเครื่องมือเพื่อช่วยวิเคราะห์สถานการณ์การแพร่ระบาดของ เชื้อโควิด-19 แสดงข้อมูลผลการทำนายรายจังหวัดในรูปแบบเว็บแอปพลิเคชัน

3.2 โครงสร้างสถาปัตยกรรมของระบบ

โครงสร้างสถาปัตยกรรมของระบบประกอบด้วยสามโมดูลหลัก คือ โมดูลที่ 1 ทำหน้าที่จัดการข้อมูล โมดูลที่ 2 เป็นกระบวนการสร้างโมเดลทำนาย โดยใช้เทคนิคการเรียนรู้เชิงลึก โมดูลที่ 3 ทำหน้าที่แสดงผลการทำนาย โดยใช้ระบบ Geographic Information System (GIS) โครงสร้างของระบบแสดงในภาพ 19



ภาพ 19 สถาปัตยกรรมโครงสร้างของระบบ

ภาพ 19 แสดงถึงระบบที่ถูกออกแบบมาเพื่อคาดการณ์สถานการณ์การแพร่ระบาดของโควิด-19 ด้วยการใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งเป็นการรวมกันของหลายโมดูลที่ทำงานร่วมกันอย่างมีประสิทธิภาพ การออกแบบระบบนี้แบ่งออกเป็นสามโมดูลหลัก ได้แก่ โมดูล 1 การจัดการข้อมูล โมดูล 2 การพัฒนาโมเดลสำหรับการทำนายและโมดูล 3 โมดูลสำหรับการจัดการเว็บแอปพลิเคชัน โดยแต่ละโมดูลมีบทบาทและหน้าที่เฉพาะเจาะจง โมดูล 1 ทำหน้าที่เกี่ยวกับการรวบรวมข้อมูลและการประมวลผลข้อมูลเบื้องต้น ข้อมูลจะดึงมาจากฐานข้อมูลของกรมควบคุมโรค (MoPH's Database) ซึ่งเก็บข้อมูลเกี่ยวกับสุขภาพและข้อมูลการแพร่ระบาดของโควิด-19 ข้อมูลที่รวบรวมได้จะส่งไปยังขั้นตอนการประมวลผลเพื่อทำความสะอาดและเตรียมความพร้อมสำหรับการนำไปใช้ในโมเดลการทำนาย การประมวลผลนี้รวมถึงการจัดการกับข้อมูลที่หายไป การปรับรูปแบบข้อมูล และการเลือกคุณสมบัติที่เหมาะสม โมดูล 2 เป็นหัวใจของระบบ โดยใช้เทคนิคการเรียนรู้เชิงลึกในการทำนายผลลัพธ์ ประกอบด้วยสองโมเดลหลักคือ LSTM และ MLP โมเดล LSTM ได้ออกแบบมาเพื่อ

จัดการกับข้อมูลที่มีความสัมพันธ์ในลำดับเวลา เช่น ข้อมูลการแพร่ระบาดรายวันหรือรายเดือน ขณะที่ MLP จะช่วยในการจัดการกับข้อมูลที่มีโครงสร้างที่ไม่ซับซ้อน ข้อมูลที่ผ่านการประมวลผลจากโมดูล 1 จะส่งเข้ามาในทั้งสองโมเดลนี้ และผลลัพธ์ที่ได้จะนำมารวมกัน (Concatenate) ผ่านเลเยอร์ Dense สุดท้ายเพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำ โมดูล 3 เป็นส่วนที่แสดงผลลัพธ์ของการทำนายโดยนำผลลัพธ์จากโมดูล 2 มาประมวลผลและแสดงผลในรูปแบบที่เข้าใจง่าย เช่น แผนที่ กราฟ และข้อมูลเชิงวิเคราะห์ต่างๆ ส่วนนี้มีความสำคัญมากเนื่องจากช่วยให้ผู้ใช้สามารถนำข้อมูลไปใช้ในการวางแผนและตัดสินใจได้อย่างมีประสิทธิภาพ การแสดงผลที่ชัดเจนและช่วยให้การตัดสินใจที่ดีขึ้นในการจัดการและควบคุมการแพร่ระบาดของโควิด-19

3.3 โมดูลที่ 1 การจัดการข้อมูล

ในการจัดการข้อมูลทั้งหมดของงานวิจัยนี้ โมดูลนี้ดึงข้อมูลมาจากฐานข้อมูลของกรมควบคุมโรคติดต่อ ผ่าน API ที่ได้รับจากรัฐบาลของประเทศไทย โดยข้อมูลที่ดึงมาใช้นั้นในการพัฒนาโมเดลสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 นั้นจะประกอบไปด้วย จำนวนผู้ติดเชื้อใหม่ จำนวนผู้ติดเชื้อสะสม จำนวนผู้เสียชีวิตใหม่ และจำนวนผู้เสียชีวิตสะสม โดยโมดูลนี้จะมีหน้าที่จัดการเกี่ยวกับชุดข้อมูลทั้งหมดเพื่อเตรียมเข้าสู่การประมวลผลหรือการฝึกอบรมของโมเดล โดยโมดูลที่ 1 นี้จะประกอบไปด้วยสองขั้นตอนหลักได้แก่ การรวบรวมข้อมูล และการประมวลผลข้อมูลล่วงหน้า

3.3.1 การเก็บรวบรวมข้อมูล

สำหรับการเก็บรวบรวมข้อมูลเพื่อสำหรับการนำไปใช้เป็นชุดข้อมูลสำหรับการทำลองการพัฒนาโมเดลสำหรับการทำนายนั้น ผู้วิจัยได้ทำการดึงข้อมูลผ่าน API จากเว็บไซต์ https://covid19.ddc.moph.go.th/ ซึ่งเป็นเว็บไซต์จากกรรมควบคุมโรคของประเทศไทยและเป็นแหล่งข้อมูลที่เชื่อถือได้สำหรับการนำมาใช้เป็นชุดข้อมูลในการพัฒนาโมเดลในครั้งนี้ โดยข้อมูลภายในโครงสร้างของ API จะประกอบไปด้วย วัน/เดือน/ปี จังหวัด จำนวนผู้ติดเชื้อใหม่ จำนวนผู้ติดเชื้อสะสม จำนวนผู้เสียชีวิตใหม่ และจำนวนผู้เสียชีวิตสะสม โดยข้อมูลที่จะเลือกใช้ในงานวิจัยนี้จะเลือกตั้งแต่วันที่มีการยืนยันการแพร่ระบาดของเชื้อโควิด-19 ในประเทศไทยคือวันที่ 12 มกราคม 2563 จนถึง 20 กุมภาพันธ์ 2566(เป็นวันสุดท้ายของการเผยแพร่มูลในรูปแบบ วัน/เดือน/ปี) ลิงค์ของ AIP ที่ใช้ทำการดึงข้อมูล คือ

https://covid19.ddc.moph.go.th/api/Cases/round-1to2-by-provinces

https://covid19.ddc.moph.go.th/api/Cases/report-round-3-y21-line-lists-by-province

ข้อมูลที่ดึงมาจาก API นั้นจะทำการจัดเก็บแยกเป็นของแต่ละจังหวัด ทั้งหมด 77 จังหวัดในรูปแบบของไฟล์ .csv ประกอบไปด้วยข้อมูลจำนวน 7 คอลัมน์ดังนี้

- txn_date คือ วัน เดือน ปี
- Province คือ จังหวัด
- new_case คือ จำนวนผู้ติดเชื้อใหม่
- new_case_excludeabroad คือ จำนวนผู้ติดเชื้อใหม่แบบนับรวม
- total_case_excludeabroad คือ จำนวนผู้ติดเชื้อสะสม
- new_death คือ จำนวนผู้เสียชีวิตใหม่
- total_death คือ จำนวนผู้เสียชีวิตสะสม

ตัวอย่างข้อมูลที่ดึงมาจาก API ของกรมควบคุมโรค ข้อมูลเริ่มตั้งแต่วันที่ 12 มกราคม 2563 จนถึงวันที่ 20 กุมภาพันธ์ 2566 แสดงในภาพ 20

	txn_date	province	new_case	new_case_excludeabroad	total_case_excludeabroad	new_death	total_death
0	1-12-2020	ทั้งประเทศ	1	1	0	0	0
1	1-12-2020	พิจิตร	0	0	0	0	0
2	1-12-2020	หนองคาย	0	0	0	0	0
3	1-12-2020	สมุทรสาคร	0	0	0	0	0
4	1-12-2020	อำนาจเจริญ	0	0	0	0	0
...	...	...	...	...	...	...	...
89739	2-20-2023	สุรินทร์	0	53393	53371	0	221
89740	2-20-2023	มุกดาหาร	0	9624	9618	0	67
89741	2-20-2023	นครพนม	0	19435	19433	0	112
89742	2-20-2023	ชลบุรี	0	237645	234640	0	1366
89743	2-20-2023	สงขลา	0	7206	7205	0	75

89744 rows × 7 columns

ภาพ 20 ข้อมูลที่ได้จาก API ในรูปแบบของกรอบข้อมูล

3.3.2 การประมวลผลข้อมูลล่วงหน้า

เมื่อได้ข้อมูลดิบจากการดึงข้อมูลผ่านทาง API และจัดเก็บแยกชุดข้อมูลของแต่ละจังหวัดแล้วนั้น ชุดข้อมูลทั้ง 77 ชุดจะต้องนำเข้าสู่กระบวนการการประมวลผลข้อมูลล่วงหน้าเป็นขั้นตอนก่อนที่จะนำข้อมูลเข้าสู่การประมวลผลภายในโมเดล โดยขั้นตอนนี้มีความสำคัญเป็นอย่างมากต่อประสิทธิภาพความแม่นยำของโมเดล โดยข้อมูลที่ใช้จะอยู่ในรูปแบบของ อนุกรมเวลาดังนั้นจึง

จำเป็นต้องทำการพิสูจน์ให้แน่ใจว่าข้อมูลมีการเรียงลำดับขั้นของเวลาที่ถูกต้องและสัมพันธ์กันระหว่างข้อมูลอื่น เช่น จำนวนผู้ติดเชื้อสะสม จำนวนผู้เสียชีวิตสะสมหรือไม่ การประมวลผลข้อมูลล่วงหน้าประกอบไปด้วยสี่ขั้นตอนดังนี้

1) ขั้นตอนการแยกข้อมูล

เป็นขั้นตอนของการแยกข้อมูลต่างๆออกจากเฟรมเดียวกันเพื่อให้สามารถทำการจัดการข้อมูลของแต่ละจังหวัดได้ง่ายขึ้นและเพื่อลดระยะเวลาในการอ่านเฟรมข้อมูลรวมไปถึงลดทรัพยากรที่ต้องใช้ในการประมวลผลเฟรมข้อมูลที่มีขนาดใหญ่ในทุกครั้งของการประมวล โดยสามารถใช้คำสั่งวนซ้ำเพื่อทำการแยกข้อมูลของแต่ละจังหวัดและทำการบันทึกไฟล์ในรูปแบบ .csv ดังภาพ 21 และตัวอย่างข้อมูลแสดงในภาพ 22

ภาพ 21 ขั้นตอนการแยกข้อมูลของแต่ละจังหวัดจากกรอบข้อมูล

	txn_date	new_case	total_case_excludeabroad	new_death	total_death
0	2020-01-12	1	0	0	0
1	2020-01-13	0	0	0	0
2	2020-01-14	0	0	0	0
3	2020-01-15	0	0	0	0
4	2020-01-16	0	0	0	0
...	...	...	...	...	...
1131	2023-02-16	5	983083	0	8470
1132	2023-02-17	5	983088	0	8470
1133	2023-02-18	8	983096	0	8470
1134	2023-02-19	0	983096	0	8470
1135	2023-02-20	0	983096	0	8470

1136 rows × 5 columns

ภาพ 22 ตัวอย่างการแสดงข้อมูลของจังหวัดกรุงเทพมหานครโดยระบุแถวและคอลัมน์

ภาพ 22 เป็นภาพของข้อมูลที่แสดงสถิติของโควิด-19 ในกรุงเทพมหานคร โดยประกอบด้วยข้อมูลที่สำคัญ ได้แก่ วันที่ (txn_date) จำนวนผู้ป่วยใหม่ในแต่ละวัน (new_case) จำนวนติดเชื้อสะสมภายในประเทศ (total_case_excludeabroad) จำนวนผู้เสียชีวิตใหม่ในแต่ละวัน (new_death) และจำนวนผู้เสียชีวิตสะสม (total_death) ข้อมูลในภาพ 22 ครอบคลุมช่วงเวลาตั้งแต่วันที่ 12 มกราคม 2020 จนถึงวันที่ 20 กุมภาพันธ์ 2023 รวมทั้งหมด 1136 วัน เมื่อรวมข้อมูลของทั้ง 77 จังหวัด มีข้อมูลทั้งหมด 87,472 ข้อมูล ภาพ 22 ช่วยให้เห็นถึงการแพร่ระบาดของโควิด-19 ในกรุงเทพมหานคร รวมถึงจำนวนผู้ป่วยและผู้เสียชีวิตที่เกิดขึ้นในแต่ละวัน ซึ่งเป็นประโยชน์ในการติดตามและวิเคราะห์สถานการณ์การแพร่ระบาดได้อย่างมีประสิทธิภาพ

	File Name	Rows	Columns
0	Samut Songkhram.csv	1136	5
1	Phayao.csv	1136	5
2	Chanthaburi.csv	1136	5
3	Surin.csv	1136	5
4	Phra Nakhon Si Ayutthaya.csv	1136	5
...	...	...	...
73	Chaiyaphum.csv	1136	5
74	Bueng Kan.csv	1136	5
75	Kanchanaburi.csv	1136	5
76	Trang.csv	1136	5
77	summary.csv	77	3

78 rows × 3 columns

ภาพ 23 ความยาวของข้อมูลในแต่ละจังหวัด

การรวมสถิติของโควิด-19 จากทุกจังหวัดในประเทศไทย โดยแต่ละไฟล์ข้อมูลของจังหวัดมี 1136 แถว และ 5 คอลัมน์แสดงในภาพ 23 ซึ่งประกอบด้วยวันที่ จำนวนผู้ป่วยใหม่ จำนวนผู้ป่วยสะสม จำนวนผู้เสียชีวิตใหม่ และจำนวนผู้เสียชีวิตสะสม การรวมข้อมูลนี้ช่วยให้สามารถเปรียบเทียบและวิเคราะห์สถานการณ์การแพร่ระบาดของโควิด-19 ในแต่ละจังหวัดได้อย่างมีประสิทธิภาพ ข้อมูลนี้ยังเป็นการสังเกตการณ์เบื้องต้นก่อนทำการประมวลผลข้อมูลล่วงหน้าเพื่อเตรียมข้อมูลสำหรับการวิเคราะห์และการสร้างโมเดลต่อไป การสังเกตข้อมูลก่อนการประมวลผลข้อมูลล่วงหน้าเป็นขั้นตอนสำคัญในการตรวจสอบความสมบูรณ์ของข้อมูล

นอกจากข้อมูลแบบอนุกรมเวลาแล้วการทดลองนี้ยังใช้ข้อมูลแบบคงที่เพื่อพัฒนาโมเดลสำหรับทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสม ตัวอย่างข้อมูลแบบคงที่แสดงในภาพ 24

	Unnamed: 0	population	dose1 people amount	dose2 people amount	dose3 people amount	dose4 people amount	total cases	total deaths
0	Amnat Charoen	328456	252418	234106	101838	6345	4190544	27208
1	Ang Thong	254019	180940	176949	108197	9555	11662493	98510
2	Bangkok	8411645	9946690	9250578	7297941	1646510	412830432	4280229
3	Bueng Kan	374306	258296	227425	60389	3493	4870212	16551
4	Buri Ram	1356674	1096976	1014683	544938	58709	26045258	76357
...	...	...	...	...	...	...	...	...
71	Udon Thani	1433871	1062620	987805	447610	37926	23581273	145920
72	Uthai Thani	288568	215130	205930	101586	9534	5516640	48792
73	Uttaradit	401527	319498	303124	170837	15918	5456641	52510
74	Yala	565448	363443	304835	70819	6537	29109742	194757
75	Yasothon	452793	346800	325638	190701	7192	7866845	57517

76 rows × 8 columns

ภาพ 24 ตัวอย่างข้อมูลแบบคงที่ในการประมวลผลโมเดล MLP

ภาพ 24 แสดงสถิติแบบคงที่ของจำนวนประชากรและการฉีดวัคซีนโควิด-19 ในแต่ละจังหวัดของประเทศไทย ซึ่งประกอบด้วย 76 แถว และ 8 คอลัมน์ ข้อมูลในแต่ละคอลัมน์ได้แก่ จำนวนประชากร (Population) จำนวนผู้ได้รับวัคซีนเข็มที่ 1 (Dose1 People Amount) จำนวนผู้ได้รับวัคซีนเข็มที่ 2 (Dose2 People Amount) จำนวนผู้ได้รับวัคซีนเข็มที่ 3 (Dose3 People Amount) จำนวนผู้ได้รับวัคซีนเข็มที่ 4 (Dose4 People Amount) จำนวนผู้ป่วยสะสม (Total Cases) และจำนวนผู้เสียชีวิตสะสม (Total Deaths) ข้อมูลนี้จะนำไปใช้ในการประมวลผลด้วยโมเดล MLP เพื่อตรวจสอบและวิเคราะห์รูปแบบต่างๆ ที่เกี่ยวข้องกับการแพร่ระบาดของโควิด-19 และประสิทธิภาพของการฉีดวัคซีนในแต่ละจังหวัด การใช้ข้อมูลสถิตินี้จะช่วยให้โมเดลสามารถเรียนรู้ถึงปัจจัยที่มีผลต่อสถิติของการแพร่ระบาดของเชื้อโควิด-19 ได้ครอบคลุมขึ้น

2) ขั้นตอนการกรองข้อมูลที่ไม่ต้องการ

ขั้นตอนนี้เป็นการคัดกรองข้อมูลส่วนเกินหรือข้อมูลที่ไม่ต้องการใช้ออก เพื่อให้ชุดข้อมูลมีความถูกต้องมากที่สุดสำหรับการนำชุดข้อมูลไปฝึกฝนในโมเดลเพื่อลดอัตราการประมวลผลข้อมูลที่ผิดพลาด ในการลดข้อมูลส่วนเกิดออกนั้นได้ใช้คำสั่ง Drop ในภาษา Python เพื่อทำการลบข้อมูลที่ไม่ต้องการออก โดยจากการประมวลผลการกรองข้อมูลในครั้งนี้จะคัดกรองข้อมูลที่ไม่จำเป็นออกได้แก่ “ไม่ระบุ” และ “ทั้งประเทศ” ดังภาพ 25

	txn_date	province	new_case	new_case_excludeabroad	total_case_excludeabroad	new_death	total_death
55	2020-01-12	ไม่ระบุ	0	0	0	0	0
141	2020-01-13	ไม่ระบุ	0	0	0	0	0
194	2020-01-14	ไม่ระบุ	0	0	0	0	0
308	2020-01-15	ไม่ระบุ	0	0	0	0	0
377	2020-01-16	ไม่ระบุ	0	0	0	0	0
...	...	...	...	...	...	...	...
89359	2023-02-16	ไม่ระบุ	0	246	234	0	0
89480	2023-02-17	ไม่ระบุ	0	246	234	0	0
89513	2023-02-18	ไม่ระบุ	0	246	234	0	0
89648	2023-02-19	ไม่ระบุ	0	246	234	0	0
89697	2023-02-20	ไม่ระบุ	0	246	234	0	0

1136 rows × 7 columns

ภาพ 25 ข้อมูลที่ต้องการกรองออก

3) การจัดเรียงอนุกรมเวลา

ขั้นตอนการจัดเรียงข้อมูลแบบอนุกรมเวลาเป็นขั้นตอนหนึ่งที่มีความสำคัญมากเนื่องจากข้อมูลที่ใช้นั้นจำเป็นต้องอยู่ในรูปแบบชุดข้อมูลที่มีลำดับและความเกี่ยวข้องกันต่อเนื่องตามลำดับของเวลาเพื่อให้สอดคล้องกับความต้องการข้อมูลของ LSTM ที่ต้องการข้อมูลที่เป็นลำดับขั้นเช่น อนุกรมเวลา โดยการเรียงข้อมูลเวลาให้มีความถูกต้องนั้นสามารถใช้คำสั่ง parse_dates ของชุดคำสั่ง Datetime ของภาษา python ได้ โดยการเลือกที่คอลลัม txn_date เพื่อทำการจัดเรียง

4) การปรับระดับของข้อมูลเพื่อลดความห่างของข้อมูล

ข้อมูลที่ผ่านขั้นตอนจัดเรียงข้อมูลแบบอนุกรมเวลาอาจะมีระยะห่างของข้อมูลที่มาก อาจะส่งผลให้การประมวลผลของโมเดลเกิดการ Bias ขึ้นในอนาคต ระยะห่างของข้อมูลแสดงในภาพ 26 แสดงให้เห็นถึงจำนวนผู้ติดเชื้อสะสมที่แสดงในคอลัมน์ total_case_excludeaboard ที่มีระยะห่างของข้อมูลตั้งแต่ 0 ถึง 983,996 ที่จำเป็นจะต้องปรับลดความห่างของข้อมูลนี้

	txn_date	new_case	total_case_excludeabroad	new_death	total_death
0	2020-01-12	1	0	0	0
1	2020-01-13	0	0	0	0
2	2020-01-14	0	0	0	0
3	2020-01-15	0	0	0	0
4	2020-01-16	0	0	0	0
...	...	...	...	...	...
1131	2023-02-16	5	983083	0	8470
1132	2023-02-17	5	983088	0	8470
1133	2023-02-18	8	983096	0	8470
1134	2023-02-19	0	983096	0	8470
1135	2023-02-20	0	983096	0	8470

1136 rows × 5 columns

ภาพ 26 ข้อมูลหลังจากการจัดเรียงวันที่ตามลำดับเวลา

การปรับระดับของข้อมูลปรับสำหรับการลดระยะหรือลดความต่างกันของข้อมูลเพื่อไม่ให้ข้อมูลหรือคุณลักษณะเด่นที่จำเป็นต้องใช้นั้นมีความต่างกันมากเกินไปและเพื่อลดการเลือกน้ำหนักที่ผิดพลาดของโมเดลอันเป็นผลไปสู่การทำนายที่ผิดพลาด ในขั้นตอนนี้ได้ใช้วิธีการทำการ Normalization [111, 112] (ปรับค่าให้อยู่ในช่วงที่กำหนด) ข้อมูลในการเตรียมข้อมูลเพื่อนำไปใช้ในกระบวนการเรียนรู้ของโมเดล Machine Learning หรือการเรียนรู้เชิงลึก โดยทำการปรับค่าข้อมูลให้เป็นช่วงค่าที่กำหนด (ตั้งแต่ -1 ถึง 1 เป็นต้น) เพื่อให้การเรียนรู้ของโมเดลมีประสิทธิภาพสูงขึ้น ซึ่งค่าข้อมูลที่ได้จากการ Normalization จะอยู่ในช่วงที่กำหนดและไม่เกินช่วงที่กำหนด โดยมีวิธีการดังนี้

หาค่าต่ำสุด (Min) และค่าสูงสุด (Max) ของข้อมูลในแต่ละคอลัมน์ที่ต้องการทำการ Normalization ใช้สูตรดังนี้ในการ Normalization ค่าข้อมูลในแต่ละคอลัมน์ในสมการที่ (7)

$$\text{ค่าข้อมูลใหม่} = (\text{ค่าข้อมูลเดิม} - \text{ค่าต่ำสุด}) / (\text{ค่าสูงสุด} - \text{ค่าต่ำสุด}) \quad (7)$$

โดยที่ค่าข้อมูลเดิมคือค่าข้อมูลก่อนที่จะทำการ Normalization ค่าต่ำสุดคือค่าน้อยที่สุดของข้อมูลในคอลัมน์นั้น และค่าสูงสุดคือค่ามากที่สุดของข้อมูลในคอลัมน์นั้น

3.4 โมดูลที่ 2 การพัฒนาโมเดลสำหรับการทำนาย

โมดูลที่ 2 นี้มีหน้าที่ในการจัดการกระบวนการพัฒนาโมเดลเพื่อการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 โดยเป็นการดึงข้อมูลมาจากโมดูลที่ 1 หลังจากที่ข้อมูลได้ผ่านกระบวนการการประมวลผลข้อมูลล่วงหน้าแล้วข้อมูลทั้งหมดจะจัดเตรียมเข้าสู่กระบวนการฝึกอบรมของโมเดล ในงานวิจัยนี้ได้ใช้โมเดลสองแบบ คือ LSTM และ MLP เพื่อจัดการกับข้อมูลที่เป็น Dynamic Time Series และ Static Data ตามลำดับ โมเดล LSTM สร้างขึ้นเพื่อจัดการกับข้อมูลที่เป็นอนุกรมเวลา (Dynamic Time Series) ในขณะที่โมเดล MLP สร้างขึ้นเพื่อจัดการกับข้อมูลที่มีรูปแบบคงที่หลังจากการประมวลผลข้อมูลด้วยทั้งสองโมเดลแล้ว ผลลัพธ์ที่ได้จากทั้งสองโมเดลจะนำมารวมกันเพื่อสร้างเอาท์พุตสุดท้าย และเมื่อได้ผลลัพธ์ที่สมบูรณ์แล้วจะนำไปแสดงผลด้วยกราฟบนโมดูลที่ 3 ในรูปแบบของเว็บแอปพลิเคชัน

คอมพิวเตอร์ที่ใช้ในการพัฒนา Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz แรม 8.00 GB (7.85 GB usable) graphic card RTX 2060 ภาษาที่ใช้ในการพัฒนา Python

3.4.1 การเตรียมข้อมูลและการประเมินประสิทธิภาพ

ในขั้นตอนของการเตรียมข้อมูได้ใช้ข้อมูลการติดเชื้อสะสมและการเสียชีวิตสะสมในการแบ่งชุดข้อมูลเพื่ออบรมนั้นจากข้อมูลทั้งหมดนั้น ได้แบ่งออกเป็น 2 ชุดในอัตราส่วนคือ 80 ต่อ 20 โดยร้อยละ 80 จะนำไปใช้ในการฝึกอบรมโมเดล โดยภายในร้อยละ 80 ของชุดข้อมูลสำหรับการฝึกนั้นจะถูกแบบออกเพื่อการทำ Validation อีกร้อยละ 10 ในส่วนของ ร้อยละ20 ที่แบ่งแยกไว้นั้นจะใช้ในการนำมาทดสอบโมเดลดังตาราง 4

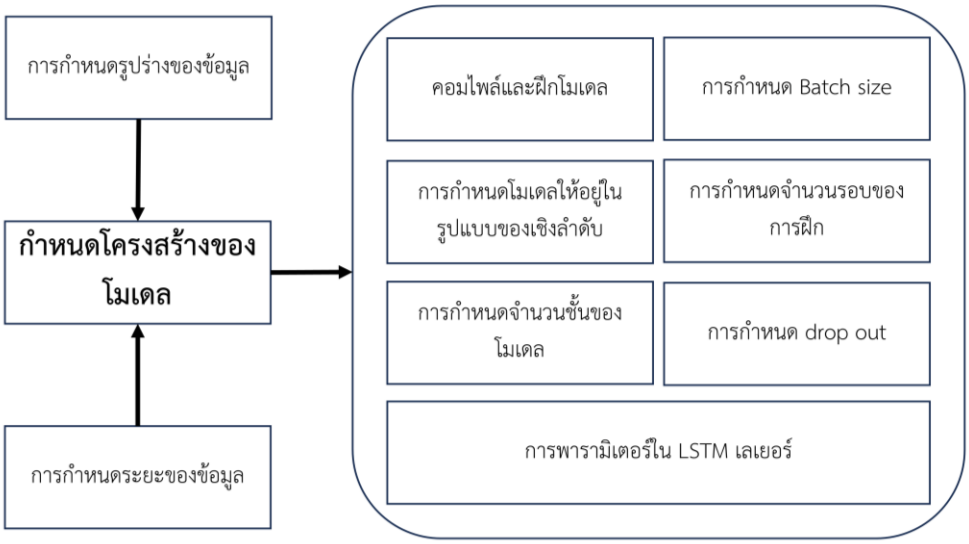
ตาราง 4 การแบ่งจำนวนข้อมูลสำหรับการพัฒนาโมเดล

การแบ่งข้อมูลที่ใช้ในการพัฒนาโมเดล	ชุดฝึกอบรม		ชุดทดสอบ
	ชุดฝึกอบรม	ชุดตรวจสอบ	
ร้อยละการแบ่งข้อมูล	70%	10%	20%

3.4.2 โมเดล LSTM แบบพื้นฐาน

การทำหนดโครงสร้างการทำงานของโมเดลประกอบไปด้วยส่วนประกอบย่อยหลายๆส่วนที่จำเป็นต้องมีการกำหนดค่าอย่างเหมาะสมเพื่อให้โมเดลมีความถูกต้องและแม่นยำ โดยโครงสร้างที่จำเป็นต้องกำหนดเพื่อสร้างโมเดล LSTM ประกอบด้วย การคอมไพล์โมเดล การกำหนดโมเดลให้อยู่ในรูปแบบเชิงลำดับ การกำหนดจำนวนชั้นของโมเดล การกไหนด Activation Function การกำหนด

Regularization การกำหนด Optimizer การกำหนด Drop Out การกำหนด Loss Function การกำหนดจำนวนรอบของการฝึกและการกำหนด Batch Size ดังภาพ 27



ภาพ 27 ส่วนประกอบการกำหนดโครงสร้างของโมเดล

การคอมไพล์และฝึกโมเดลเป็นขั้นตอนสำคัญและสำคัญอย่างมากในกระบวนการสร้างและใช้งานโมเดล LSTM ที่มีความซับซ้อนในการทำงานของโมเดลเพื่อให้โมเดลนั้นมีการทำงานที่สอดคล้องกับข้อมูลที่นำมาใช้ในการพัฒนาโมเดล การคอมไพล์โมเดลเป็นขั้นตอนที่กำหนดวิธีการเรียนรู้และอัลกอริทึมในการปรับค่า Weight ของโมเดล ซึ่งส่งผลต่อประสิทธิภาพในการเรียนรู้และทำนายของโมเดล การเลือกอัลกอริทึมที่เหมาะสมสำหรับข้อมูลที่ต้องการทำนายนั้นมีความสำคัญ นอกจากนี้ยังต้องกำหนดตัววัด (Loss Function) ที่เหมาะสมในการประเมินความผิดพลาดของโมเดล ซึ่งเป็นปัจจัยสำคัญในกระบวนการฝึกโมเดล

การสร้างโมเดล LSTM เชิงลำดับเริ่มต้นด้วยการเตรียมข้อมูลลำดับ ซึ่งหมายถึงการจัดข้อมูลให้เป็นชุดลำดับที่มีความต่อเนื่อง เช่น อนุกรมเวลา ก่อนที่จะใช้ข้อมูลเหล่านี้ในการฝึกอบรมโมเดล LSTM จะต้องกำหนดโครงสร้างที่เหมาะสมและเป็นการตั้งค่าเพื่อให้โมเดลสามารถเรียนรู้ลักษณะและความสัมพันธ์ในลำดับข้อมูลได้อย่างมีประสิทธิภาพ ข้อมูลที่ใช้ในการทดลองนี้เป็นข้อเป็นข้อมูลที่มีความซับซ้อนจึงและเป็นลำดับ (Sequential) จึงเหมาะสมกับโมเดล LSTM ที่สามารถจัดการข้อมูลแบบเป็นลำดับได้ ในการสร้างโมเดลที่ใช้กับโครงสร้างข้อมูลแบบเป็นลำดับด้วยไลบรารี Keras และ

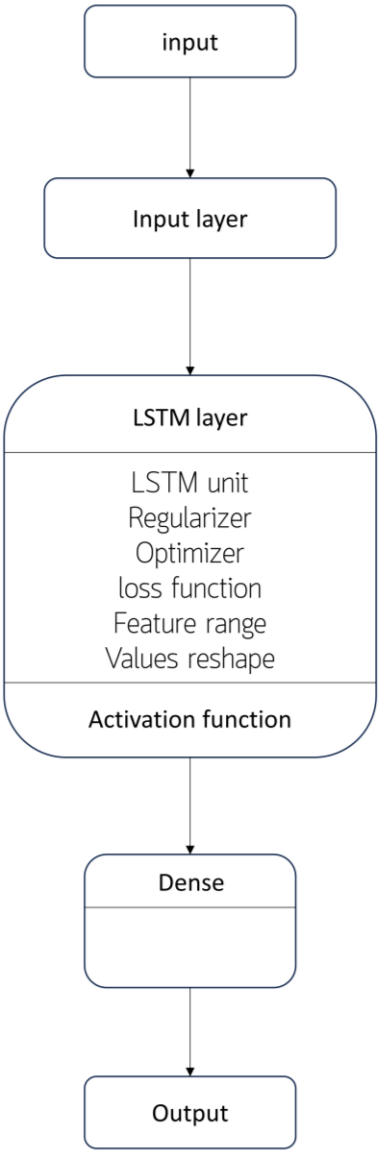
Tensor Flow จะกำหนดโครงสร้างโมเดลเป็น Model = Sequential การทดลองนี้จะกำหนดชั้นข้อมูลขึ้นอย่างเป็นลำดับตามลำดับเวลา

การทำ Dropout [113] คือ เทคนิคหนึ่งในการป้องกันการเกิด Overfitting ในโมเดล Neural Network หรือการเรียนรู้เชิงลึก โดยการใช้ Dropout เป็นการสุ่มปิด (Drop) บางส่วนของโหนดหรือหน่วยย่อยต่างๆใน Hidden Layers ขณะที่กำลังฝึกโมเดล

Batch Size คือพารามิเตอร์ที่กำหนดจำนวนตัวอย่างของชุดข้อมูลสำหรับการฝึกฝนที่จะนำมาใช้ในการอัปเดตพารามิเตอร์ภายในของโมเดล โดยการกำหนดค่าของ Batch Size นั้นมีความสำคัญเป็นอย่างมากในการให้โมเดลมีประสิทธิภาพของโมและยังส่งผลไปถึงการใช้ทรัพยากรในการประมวลผลอีกด้วย

การกำหนดจำนวนรอบของการฝึกฝนของโมเดล (Epoch) นั้นไม่มีการกำหนดที่แน่ชัดและไม่ตายตัวเนื่องจากการพัฒนาโมเดลต่างๆจำเป็นต้องมีการทดลองการกำหนดจำนวนรอบที่เหมาะสมข้อมูลที่มีอยู่ซึ่งจะมีความสัมพันธ์กับการฝึกฝนของโมเดลโดยตรง ในแต่ละ Epoch โมเดลจะได้รับข้อมูลฝึกทั้งหมดและทำการอัปเดตพารามิเตอร์ [114] เพื่อปรับให้โมเดลมีความสามารถในการแยกแยะหรือทำนายข้อมูลที่ดีขึ้น

โครงสร้างโมเดล LSTM แบบพื้นฐานซึ่งจะไปกอบไปด้วย LSTM เลเยอร์ทั้งหมดหนึ่งเลเยอร์เท่านั้น โดยภายในโมเดล LSTM เลเยอร์นั้นจะประกอบไปด้วยพารามิเตอร์ต่างๆที่ส่งผลต่อการทำงานของโมเดล LSTM แบบพื้นฐานและส่งผลต่อการทำนายโดยตรง ดังนั้นจึงต้องมีการปรับแต่งเพื่อหาผลลัพธ์พารามอเตอร์ที่ดีที่สุดของโมเดล LSTM แบบพื้นฐานในขั้นตอนถัดไป พารามิเตอร์ภายใน LSTM เลเยอร์แสดงในภาพ 28



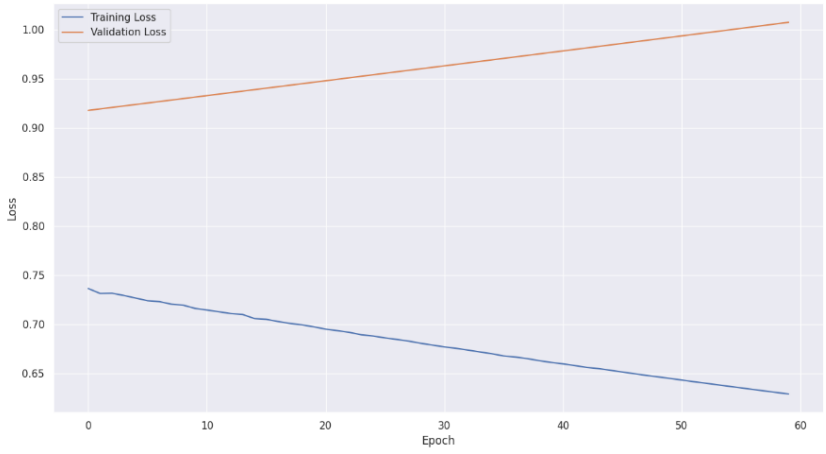
ภาพ 28 โครงสร้างของโมเดล LSTM แบบพื้นฐาน

ภาพ 28 โครงสร้างโมเดล LSTM แบบพื้นฐานซึ่งจะไปกอบไปด้วย LSTM เลเยอร์ทั้งหมดหนึ่งเลแยอร์ โดยภายใน LSTM เลเยอร์ประกอบไปด้วยพารามิเตอร์คือ 1)หน่วยของ LSTM 2) Regularizer 3) Optimizer 4) Loss Function 5) Feature Range 6) Values Reshape และ 7) Activation Function พารามิเตอร์เหล่านี้ส่งผลต่อประสิทธิภาพการทำนายของโมเดล LSTM แบบพื้นฐาน จึงต้องมีการปรับแต่งเพื่อหาค่าพารามิเตอร์ที่ดีที่สุดของโมเดล การกำหนดพารามิเตอร์ภายใน LSTM เลเยอร์มีรายละเอียดดังนี้

1) หน่วยของ LSTM

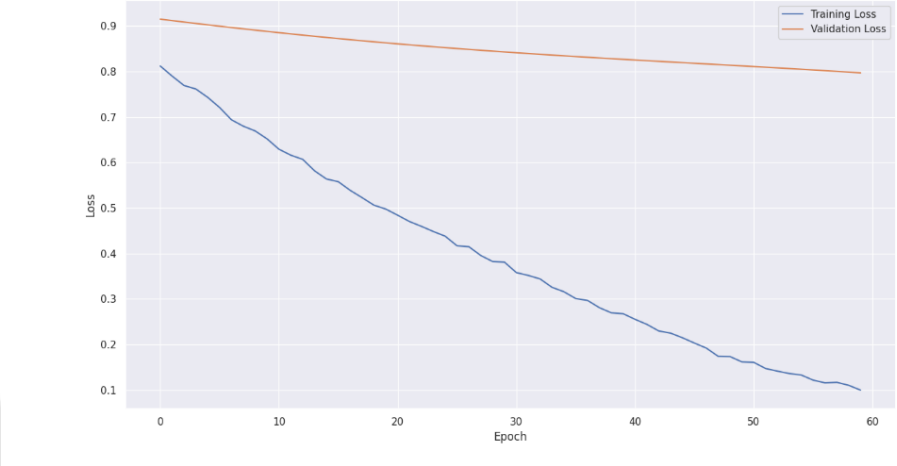
หน่วยของ LSTM คือการกำหนดจำนวนหน่วยของโมเดล LSTM ในการหาจำนวนหน่วยของโมเดลที่ดีที่สุดผู้วิจัยได้ออกแบบการทดลองโดยการกำหนด LSTM เลเยอร์เพียงชั้นเดียวในการประมวลผลข้อมูล ซึ่งจำนวนหน่วยที่ใช้มากขึ้นอาจทำให้โมเดลมีมีประสิทธิภาพมากขึ้นแต่จำนวนหน่วยที่มากเกินไปอาจส่งผลต่อการใช้ทรัพยากรในการประมวลผลมากเกินไปและอาจจะเสี่ยงต่อการเกิด Overfitting ดังนั้นจึงจำเป็นต้องหาจำนวนหน่วยที่เหมาะสม โดยการทดลองการปรับค่าหน่วยของ LSTM เลเยอร์จะทำการปรับตั้งแต่หน่วยของ LSTM ที่ 100 จนถึง 700 โดยผลลัพธ์ของการปรับหน่วยแสดงดังต่อไปนี้

ภาพ 29 กราฟนี้แสดงผลการฝึกโมเดล LSTM ซึ่งได้ตั้งค่าให้ใช้ 100 LSTM หน่วยโดยมีการวัดค่าการสูญเสีย (Loss) ของทั้งชุดข้อมูลการฝึก (Training Loss) และชุดข้อมูลการทดสอบ (Validation Loss) ในแต่ละ Epoch เพื่อประเมินประสิทธิภาพของโมเดลในการเรียนรู้จากข้อมูล กราฟแสดงว่าในช่วงแรกของการฝึก (Epoch 0-10) ค่าการสูญเสียของชุดข้อมูลการฝึก (เส้นสีน้ำเงิน) นั้นมีแนวโน้มที่ลดลง ซึ่งบ่งชี้ว่าโมเดลกำลังเรียนรู้รูปแบบจากข้อมูลอย่างรวดเร็ว เมื่อ Epoch เพิ่มขึ้น ค่าการสูญเสียของชุดข้อมูลการฝึกยังคงลดลงอย่างต่อเนื่องจนเข้าสู่ค่าที่ต่ำลง แสดงว่าโมเดลเริ่มสามารถจับรูปแบบของข้อมูลการฝึกได้ อย่างไรก็ตาม ค่าการสูญเสียของชุดข้อมูลการทดสอบ (เส้นสีส้ม) มีการเพิ่มขึ้นซึ่งอาจบ่งชี้ถึงการเกิด Overfitting ที่ชัดเจนมาก เนื่องจากโมเดลเรียนรู้ชุดข้อมูลไม่สดคล้องกับชุดทดสอบ



ภาพ 29 Learning Curve ของโมเดล LSTM แบบพื้นฐาน แบบ 100 หน่วย

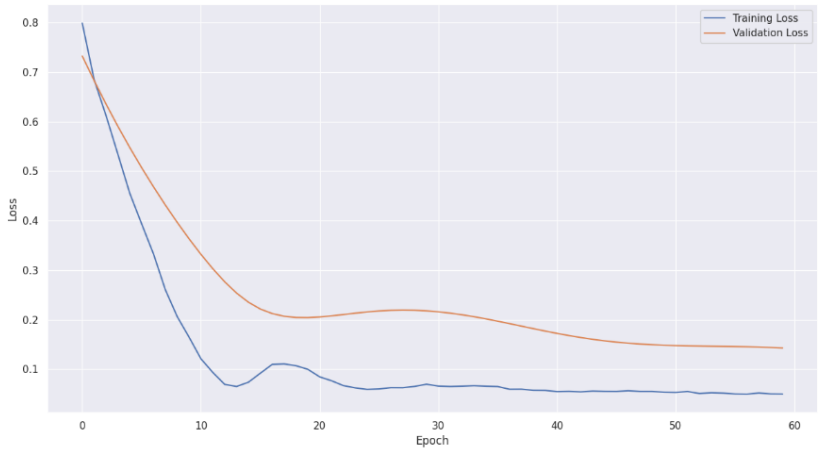
ภาพ 30 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 200 LSTM หน่วย โดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น



ภาพ 30 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 200 หน่วย

เมื่อเปรียบเทียบกับโมเดลที่ใช้ 100 LSTM หน่วย จะเห็นได้ชัดว่าโมเดลที่ใช้ 200 หน่วย แสดงประสิทธิภาพดีกว่าในหลายๆ ด้าน โดยเฉพาะอย่างยิ่งในแง่ของการลดค่าการสูญเสียของชุดข้อมูลการทดสอบ การที่เส้นของ Training Loss และ Validation Loss มีแนวโน้มเริ่มเข้าใกล้กันมากขึ้นบ่งชี้ว่าโมเดลนี้สามารถปรับตัวได้ดีขึ้นกับข้อมูลทั้งสองชุด ซึ่งเป็นสัญญาณที่ดีว่าการเพิ่มจำนวนหน่วยส่งผลให้โมเดลสามารถเรียนรู้ได้อย่างลึกซึ้งและครอบคลุมมากขึ้น ในขณะที่โมเดลที่ใช้ 100 หน่วยนั้นมีการลดลงของค่าการสูญเสียในช่วงแรกเช่นเดียวกัน แต่เมื่อ Epoch เพิ่มขึ้น เส้นของ Validation Loss มีความชันที่ช้าลงและไม่ลดลงมากเท่าที่ควร ซึ่งอาจบ่งชี้ว่าโมเดลมีความสามารถในการทำนายข้อมูลใหม่ไม่ดีเท่าที่ควร โมเดลที่ใช้ 200 หน่วยแสดงให้เห็นถึงการเรียนรู้ที่ดีขึ้นแต่ก็ยังมีแนวโน้มลดความเสี่ยงของการเกิด Overfitting

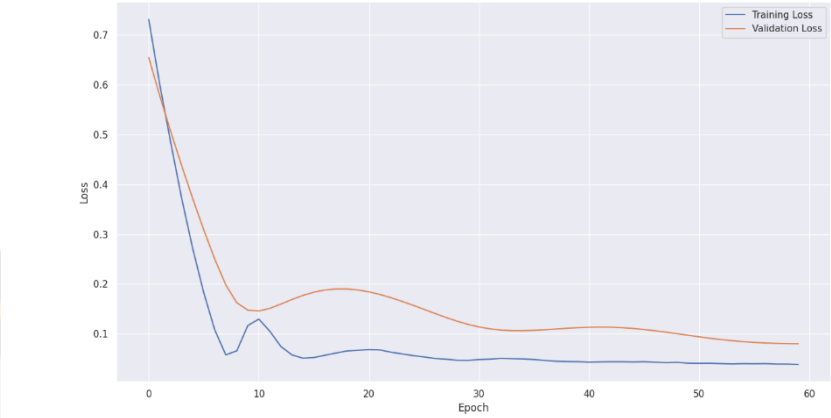
ภาพ 31 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 300 LSTM หน่วยโดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น



ภาพ 31 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 300 หน่วย

ภาพ 31 เมื่อเปรียบเทียบกราฟการเรียนรู้ของ LSTM ที่มี 200 หน่วย กับ 300 หน่วย จะเห็นได้ว่าทั้งสองโมเดลมีแนวโน้มที่ดีในการลดลงของ Training Loss แต่มีความแตกต่างกันในบางประการ สำหรับโมเดล LSTM ที่มี 200 หน่วย แสดงให้เห็นว่าโมเดลสามารถเรียนรู้จากข้อมูลการฝึกได้ระดับหนึ่ง โดย Training Loss ลดลงอย่างต่อเนื่อง ขณะที่ Validation Loss ลดลงเช่นกัน แต่ในอัตราที่ช้ากว่า ซึ่งสะท้อนถึงความสามารถของโมเดลในการทำนายข้อมูลใหม่ ในส่วนของโมเดล LSTM ที่มี 300 หน่วย แสดงให้เห็นถึงประสิทธิภาพสูงขึ้นในการเรียนรู้ โดย Training Loss ลดลงอย่างรวดเร็วและต่อเนื่อง ขณะที่ Validation Loss ก็มีแนวโน้มลดลงอย่างชัดเจน แสดงให้เห็นว่าโมเดลนี้สามารถทำงานได้ดียิ่งขึ้นในการทำนายข้อมูลใหม่ๆ และลดความเสี่ยงของการ Overfitting ซึ่งทำให้โมเดลที่มี 300 หน่วย มีประสิทธิภาพโดยรวมดีกว่าโมเดลที่มี 200 หน่วย แต่ก็ยังเกิดปัญหา Overfitting อยู่เนื่องจากข้อมูลทั้งสองเส้นนั้นยังไม่สอดคล้องกัน

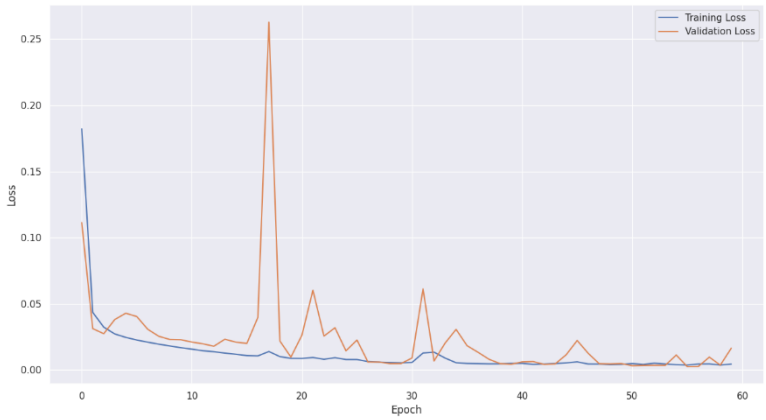
ภาพ 32 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 400 LSTM หน่วย โดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น



ภาพ 32 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 400 หน่วย

เมื่อเปรียบเทียบ LSTM ที่มี 300 หน่วยกับ 400 หน่วย พบว่าทั้งสองโมเดลสามารถลด Training Loss ได้อย่างรวดเร็วและมีเสถียรภาพ แต่โมเดลที่มี 400 หน่วย แสดงให้เห็นถึงความสามารถในการลด Training Loss ไปสู่ระดับที่ต่ำมากกว่า และมีแนวโน้มที่ Validation Loss จะลดลงและเสถียรในระยะยาว ซึ่งบ่งบอกว่าโมเดลนี้สามารถจัดการกับข้อมูลที่ซับซ้อนได้ดีกว่า โดยโมเดล 400 หน่วย สามารถจับรายละเอียดที่ซับซ้อนได้ดีขึ้น ทำให้มีความแม่นยำในการทำนายข้อมูลใหม่ๆ สูงขึ้น แม้ว่าจะมีความเสี่ยงของการ Overfitting ในช่วงแรก แต่ในระยะยาว Validation Loss กลับมีความเสถียรในระดับต่ำกว่า ในทางกลับกัน โมเดลที่มี 300 หน่วย แม้จะมี Validation Loss ที่เสถียรในระดับต่ำ แต่ไม่สามารถลดลงได้มากเท่ากับโมเดลที่มี 400 หน่วย อย่างไรก็ตามการใช้ 400 หน่วย นั้นจึงยังไม่พอสำหรับการเรียนรู้ข้อมูลเนื่องจากยังไม่สามารถกับปัญหา Overfitting ได้

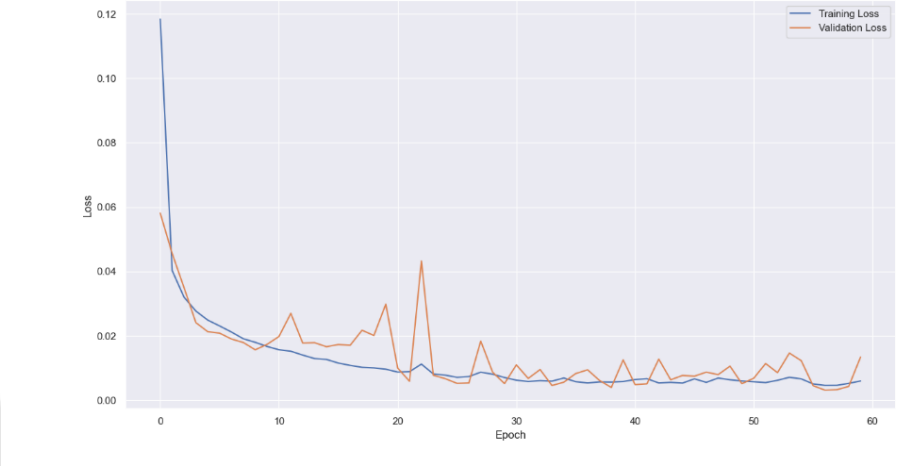
ภาพ 33 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 500 LSTM หน่วย โดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น



ภาพ 33 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 500 หน่วย

เมื่อเปรียบเทียบโมเดล LSTM ที่มี 400 หน่วย และ 500 หน่วยจะเห็นว่าโมเดลที่มี 500 หน่วย แสดงการเข้าใกล้กันของเส้น Training Loss และ Validation Loss มากขึ้น ซึ่งบ่งบอกถึงการเรียนรู้ที่มีความสอดคล้องกันระหว่างการฝึกและการทำนายข้อมูลใหม่ ความใกล้เคียงนี้เป็นสัญญาณที่ดีว่าการเพิ่มหน่วยทำให้โมเดลสามารถจับรายละเอียดของข้อมูลได้ดีขึ้น แต่หากพิจารณาจากเส้น Validation Loss บางช่วงยังมีการพุ่มขึ้นของกราฟซึ่งบ่งชี้ว่าโมเดลนี้ยังคงมีปัญหา Overfitting อยู่ โดยโมเดลสามารถเรียนรู้ข้อมูลการฝึกได้แต่ไม่สามารถทำนายข้อมูลใหม่ได้อย่างเสถียร แม้ว่าโมเดลที่มี 400 หน่วย จะมีความเสถียรในด้านของ Validation Loss มากกว่า แต่ก็ยังคงมี Overfitting โดย Validation Loss อยู่ในระดับต่ำแต่ไม่ลดลงมากนักในระยะยาว

ภาพ 34 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 600 LSTM หน่วย โดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น

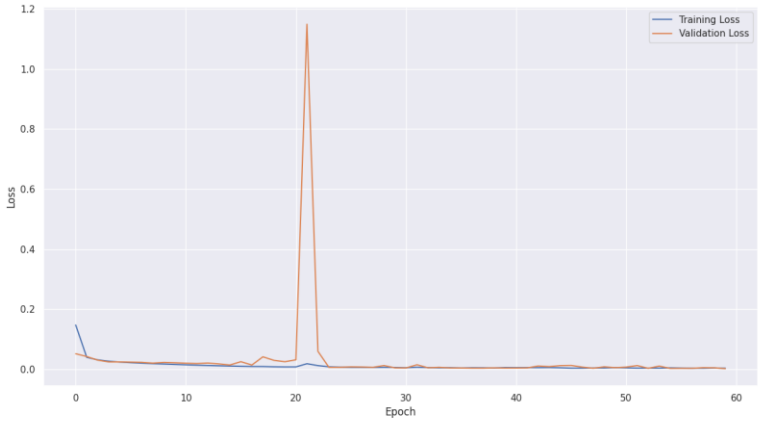


ภาพ 34 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 600 หน่วย

เมื่อเปรียบเทียบ LSTM ที่มี 500 หน่วย กับ 600 หน่วย จะพบว่าโมเดลที่มี 500 หน่วย แสดงให้เห็นถึงการลดลงของ Training Loss อย่างรวดเร็ว แต่มีปัญหาความผันผวนของ Validation Loss ซึ่งบ่งชี้ถึงความไม่เสถียรในการทำนายข้อมูลใหม่ แม้ว่าเส้นของ Training Loss และ Validation Loss จะเข้าใกล้กันมากขึ้น แต่ความผันผวนของ Validation Loss ยังคงแสดงให้เห็นถึงปัญหา Overfitting ที่ยังคงมีอยู่ ทำให้โมเดลนี้อาจไม่สามารถทำนายข้อมูลใหม่ได้อย่างแม่นยำ

โมเดลที่มี 600 หน่วยแสดงถึงความเสถียรที่ดีกว่าในแง่ของ Validation Loss แม้ว่าจะยังคงมีการแกว่งตัวอยู่ แต่ความผันผวนลดลงเมื่อเทียบกับโมเดล 500 หน่วย ซึ่งบ่งชี้ว่าโมเดลนี้สามารถจัดการกับข้อมูลได้ดีขึ้นและลดความเสี่ยงของการ Overfitting ได้ดีขึ้นเล็กน้อย การที่เส้น Validation Loss และ Training Loss เข้าใกล้กันมากขึ้นและมีความผันผวนน้อยลงทำให้โมเดลนี้ดูจะมีประสิทธิภาพในการทำนายข้อมูลใหม่ที่แม่นยำและสม่ำเสมอกว่าโมเดลที่มี 500 หน่วย

ภาพ 35 กราฟนี้แสดงผลการฝึกโมเดล LSTM ที่ใช้ 700 LSTM หน่วย โดยในแต่ละ Epoch ได้มีการวัดค่าการสูญเสียทั้งในชุดข้อมูลการฝึกและชุดข้อมูลการทดสอบ กราฟแสดงให้เห็นว่าค่าการสูญเสียของชุดข้อมูลการฝึกลดลงอย่างรวดเร็วในช่วงแรกและเข้าสู่ระดับคงที่ในระดับต่ำเมื่อ Epoch เพิ่มขึ้น



ภาพ 35 Learning Curve ของโมเดล LSTM แบบพื้นฐานแบบ 700 หน่วย

เมื่อเปรียบเทียบ LSTM ที่มี 600 หน่วย กับ 700 หน่วย จะพบว่าทั้งสองโมเดลสามารถลด Training Loss ลงได้ แต่เมื่อพิจารณา Validation Loss ซึ่งเป็นตัวบ่งชี้ความสามารถในการทำนายข้อมูลใหม่ จะเห็นได้ว่าโมเดลที่มี 700 หน่วยมีความผันผวนที่สูงขึ้นอย่างชัดเจน โดยมีจุดที่ Validation Loss เพิ่มขึ้นอย่างมาก ซึ่งเป็นสัญญาณของการเกิด Overfitting ที่ชัดเจน และมีแนวโน้มที่จะไม่สามารถทำนายข้อมูลใหม่ได้อย่างเสถียรเนื่องจากการฟิตที่มากเกินไปกับข้อมูลการฝึก โมเดลที่มี 600 หน่วย แม้จะยังคงมีการแกว่งตัวของ Validation Loss อยู่บ้าง แต่การผันผวนนั้นน้อยกว่าที่เห็นในโมเดล 700 หน่วยทำให้โมเดลที่มี 600 หน่วยดูมีความเสถียรในการทำนายข้อมูลใหม่มากกว่า ซึ่งบ่งบอกถึงการเกิด Overfitting ที่น้อยกว่า

เมื่อเปรียบเทียบกราฟการเรียนรู้จากโมเดล LSTM ที่มีจำนวนหน่วยตั้งแต่ 100 200 300 400 500 600 จนถึง 700 หน่วยจะเห็นได้ว่าแต่ละโมเดลมีลักษณะการลดลงของ Training Loss ที่ดีและมีความสามารถในการเรียนรู้ข้อมูลจากการฝึกอย่างมีประสิทธิภาพ อย่างไรก็ตาม เมื่อพิจารณา Validation Loss ซึ่งบ่งชี้ถึงความสามารถในการทำนายข้อมูลใหม่ จะพบว่าโมเดลที่มีจำนวนหน่วยน้อยกว่า 600 หน่วยมีความผันผวนของ Validation Loss ที่สูงขึ้น ซึ่งเป็นสัญญาณของการเกิด

Overfitting โดยเฉพาะในโมเดลที่มี 700 หน่วยที่แสดงถึงการผันผวนอย่างชัดเจนใน Validation Loss ในการทดลองจึงเลือกจำนวนหน่วยของ LSTM เป็น 600 หน่วย

2) การกำหนด regularization

การกำหนด regularization ในชั้น LSTM โดยการใช้ L2 Regularization [115] ซึ่งเป็นเทคนิคในการปรับค่า Weights หรือพารามิเตอร์ในโมเดลเพื่อลดความซับซ้อนและป้องกัน Overfitting [116] ในกระบวนการเรียนรู้ของในโมเดลซึ่งมีจำนวน Weights มากทำให้มีความเสี่ยงในการเกิด Overfitting ขณะฝึกโมเดลกับชุดข้อมูลฝึก

พารามิเตอร์ kernel_regularizer ในโมเดล LSTM คือการใช้ L2 Regularization เพื่อลดความเสี่ยงในการเกิด Overfitting โดยกำหนดค่า ในการคำนวณ L2 norm ของ Weights ในโมเดล LSTM ค่า ที่กำหนดสูงขึ้นจะทำให้การลดค่า weights ที่มีค่ามากขึ้นมากขึ้น นั่นหมายความว่าค่า weights จะปรับลดลงในขั้นตอนการฝึกโมเดลเพื่อลดความซับซ้อนและป้องกัน Overfitting ซึ่งช่วยให้โมเดลมีความสมดุลในการทำนายข้อมูลดีขึ้นและทำให้มีความเสถียรในการทำนายข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้ดีขึ้น การทำ L2 Regularization ในรูปแบบสมการสามารถแสดงได้ดังสมการที่ (8) [117]

$$L2\ regularization\ term = \lambda * || w ||^2 \quad (8)$$

เมื่อ

λ (Lambda) คือ พารามิเตอร์ Regularization ซึ่งกำหนดระดับของการปรับค่าน้ำหนักที่มีในโมเดล ค่า λ มากจะทำให้การ L2 Regularization หนักขึ้น และค่า λ น้อยจะทำให้การ L2 Regularization เบาลง

$|| w ||^2$ คือ ค่าระยะทางของเวคเตอร์น้ำหนัก w ในรูปของ L2 Norm (Euclidean Norm) โดยคือผลรวมของค่าน้ำหนักที่ยกกำลังสองในเวคเตอร์

การกำหนด kernel_regularizer = regularizers.l2 เท่ากับ 0.01 นั้น จะใช้ L2 Regularization ในการคำนวณ penalty [118] ของ Weights ใน LSTM เลเยอร์ โดยกำหนดค่า Lambda เท่ากับ 0.01 ที่เป็นค่าสัมประสิทธิ์ในการลดความซับซ้อนของโมเดล หลังจากนั้นในกระบวนการฝึกโมเดล ค่า weights ใน LSTM เลเยอร์จะปรับใหม่โดยลดลงเพื่อลดความซับซ้อนของโมเดลและลดความเสี่ยงในการเกิด Overfitting ของโมเดล และช่วยให้โมเดลทำนายข้อมูลได้ดีกับข้อมูลที่ไม่เคยเห็นมาก่อนและมีความเสถียรในการทำนายข้อมูลใหม่ได้ดีขึ้น

3) การกำหนด Optimizer

ในขั้นตอนของการกำหนดรูปแบบของ Optimizer นั้นได้มีการนำ Adam Optimizer [119] เข้ามาใช้ในโมเดล LSTM เพื่อกำหนดวิธีการคำนวณค่าความคลาดเคลื่อนและปรับค่าน้ำหนักของโมเดลในขั้นตอนการฝึก โดยใช้ Adam เป็นวิธีการอัลกอริทึมในการคำนวณค่าความคลาดเคลื่อนแบบเก็บค่าสุดท้าย [120] ซึ่งเป็นวิธีการที่ออกแบบมาเพื่อช่วยในกระบวนการปรับค่าน้ำหนักให้กับโมเดลให้มีความเร็วในการฝึก และให้สามารถหาค่าน้ำหนักที่เหมาะสมในการลดค่าคลาดเคลื่อนได้เร็วขึ้น

อัลกอริทึม Adam (Adaptive Moment Estimation) เป็นวิธีการปรับค่าน้ำหนักในกระบวนการฝึกโมเดลของโครงข่ายซึ่งใช้การคำนวณค่าคลาดเคลื่อนแบบเก็บค่าสุดท้าย (Momentum) และการปรับค่าความเร็วในการฝึก [121] (Learning Rate) เพื่อเพิ่มความเร็วในกระบวนการฝึกและลดปัญหาค่าต่ำ [122] (Local Minima) ที่อาจเกิดขึ้นในการค้นหาค่าน้ำหนักของโมเดลจาการคำนวณเกรเดียนต์ของฟังก์ชันสูญเสียตามค่าน้ำหนัก w ในสมการที่ (9)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{9}$$

โดยที่ m_t คือ เกรเดียนต์เครื่องหมายเวคเตอร์ ที่ t

β_1 คือ Hyperparameter สำหรับการเปรียบเทียบเกรเดียนต์เครื่องหมายปัจจุบันและเก่า

g_t คือ เกรเดียนต์ของฟังก์ชันสูญเสียตามค่าน้ำหนัก w ที่ t

คำนวณเกรเดียนต์ของฟังก์ชันสูญเสียตามค่าน้ำหนัก w ในรูปแบบของ Learning Rate ในสมการที่ (10)

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{10}$$

โดยที่ v_t คือ เกรเดียนต์เครื่องหมายเวคเตอร์ในรูปแบบเลเรทเตอร์ ที่ t

β_2 คือ Hyperparameter สำหรับการเปรียบเทียบเกรเดียนต์เครื่องหมายปัจจุบันและเก่า

g_t^2 คือ เกรเดียนต์ของฟังก์ชันสูญเสียตามค่าน้ำหนัก w ที่ t

4) การกำหนด (Loss Function)

ฟังก์ชันสูญเสีย [123] (Loss Function) เป็นฟังก์ชันที่ใช้ในการวัดระยะห่างระหว่างค่าทำนายที่ได้จากอัลกอริทึมของโมเดลกับค่าของข้อมูลออกที่เกิดขึ้นจริง (Ground Truth) ในข้อมูลการฝึกนั้นเป็นเทคนิคที่ใช้ในการประเมินว่าโมเดลทำนายข้อมูลเข้ากับข้อมูลจริงมากน้อยแค่ไหน จะใช้ในการวัดความคลาดเคลื่อนระหว่างค่าทำนายที่ได้จากโมเดลกับค่าของข้อมูลออกที่เกิดขึ้นจริงสำหรับข้อมูลการฝึกในแต่ละตัวอย่างข้อมูล ในทางตรงกันข้ามคือฟังก์ชันคอสต์ (Cost Function) [124] คือค่าเฉลี่ยของฟังก์ชันสูญเสียทั้งหมดในชุดข้อมูลการฝึกซึ่งสำหรับการฝึกโมเดลโดยใช้การแบ่งชุดข้อมูลออกเป็นชุดฝึกและชุดทดสอบ ฟังก์ชันสูญเสียจะใช้ในการคำนวณความคลาดเคลื่อนของค่าทำนายที่โมเดลทำกับชุดฝึกในขณะที่ฟังก์ชันคอสต์คือการนับความคลาดเคลื่อนเฉลี่ยของฟังก์ชันสูญเสียทั้งหมดในชุดฝึก

โดยในการพัฒนาโมเดลในครั้งนี้ ได้เลือกใช้ Loss Function ในรูปแบบของ Mean Squared Error (MSE) [125] เนื่องจากต้องการพัฒนาโมเดลที่อยู่ในรูปแบบของการทำนายและชุดข้อมูลที่จะอยู่ในรูปแบบของอนุกรมเวลาดังนั้นการเลือกใช้ MSE มาเป็น Loss Function จึงมีความเหมาะสมมากที่สุด โดยการทำงานของ MSE Loss Function สามารถอธิบายได้ด้วยสมการที่ (11)

$$MSE(y, y) = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i)^2 \tag{11}$$

เมื่อ n คือ จำนวนตัวอย่างในชุดข้อมูล

y คือ ค่าจริง

y คือ ค่าที่โมเดลทำนาย

สมการที่ (11) จะหาผลรวมของค่ากำลังสองของความต่างระหว่างค่าจริงและค่าที่ทำนายแล้วหารด้วยจำนวนตัวอย่างทั้งหมดเพื่อคำนวณค่าเฉลี่ยของความสูญเสียทั้งหมดในชุดข้อมูลการฝึก โดยที่ค่า MSE มีค่าจำกัดอยู่ในช่วง 0 ถึง ∞ แต่ค่าที่ดีที่สุดจะเป็น 0 หากค่าที่ทำนายเหมือนกับค่าจริงและจะมีค่ามากขึ้นเมื่อค่าที่ทำนายแตกต่างจากค่าจริงมากขึ้น

5) Feature Range

พารามิเตอร์นี้กำหนดช่วงของค่าที่ใช้ในการสเกลข้อมูล โดยใช้ Min-Max Scaler เพื่อแปลงค่าข้อมูลให้อยู่ในช่วงที่กำหนดไว้ -1 ถึง 1 การสเกลข้อมูลให้อยู่ในช่วงนี้ช่วยลดปัญหาค่าผลลัพธ์ที่สูง

เกินไปหรือต่ำเกินไป ซึ่งอาจทำให้โมเดลทำงานผิดพลาด การทำให้ข้อมูลอยู่ในช่วงที่สม่ำเสมอช่วยให้ฟังก์ชัน activation เช่น TanH ทำงานได้มีประสิทธิภาพมากขึ้นและช่วยให้การเรียนรู้ของโมเดลมีความเสถียรมากขึ้น

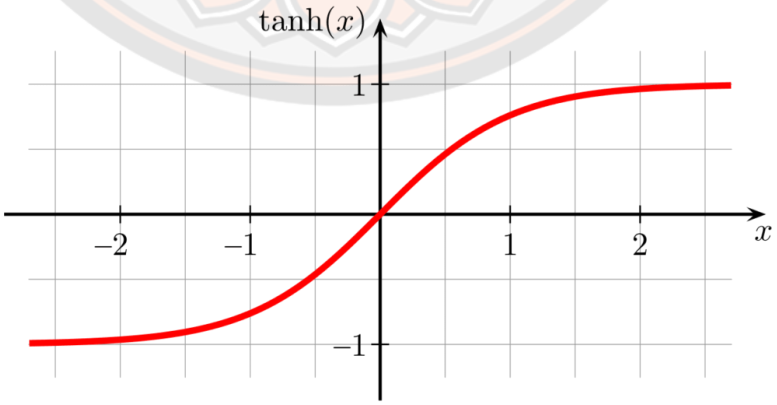
6) Values Reshape

การปรับรูปแบบข้อมูลให้เป็นเวคเตอร์ 2 มิติ (-1,1) มีความสำคัญต่อการประมวลผลข้อมูลออนุกรมเวลาในโมเดล LSTM การปรับรูปแบบนี้ช่วยให้ข้อมูลสามารถนำเข้าโมเดลได้อย่างถูกต้องและมีประสิทธิภาพ โดยเฉพาะเมื่อทำงานกับข้อมูลที่มีลำดับเวลา

7) การกำหนด Activation Function

ในการกำหนด Activation Function [126] คือการกำหนดฟังก์ชัน Activation ในชั้น LSTM โดยในการพัฒนาโมเดลครั้งนี้จะใช้ "TanH" มาเป็น Activation Function ซึ่งเป็นฟังก์ชันแทนที่ที่ใช้กำหนดการทำงานของโนเดลในแต่ละขั้นตอนของ LSTM หลังจากการประมวลผลข้อมูลในแต่ละ Step ด้วย Weights และ Biases

TanH เป็นย่อมาจาก "Hyperbolic Tangent" เป็นฟังก์ชันที่คล้ายกับฟังก์ชัน Sigmoid แต่มีช่วงค่าที่แคบกว่า อยู่ในช่วง -1 ถึง 1 ดังภาพ 36 โดยฟังก์ชัน "TanH" มีลักษณะที่คล้ายกับฟังก์ชัน Sigmoid ในระหว่าง -1 ถึง 1 และแสดงส่วนที่เน้นการกระจายค่าข้อมูลเหมาะสำหรับใช้กับโมเดล LSTM เนื่องจากมีความสามารถในการแก้ปัญหา Vanishing Gradient ที่เกิดขึ้นในการฝึกโมเดลที่มีความลึกมากขึ้น

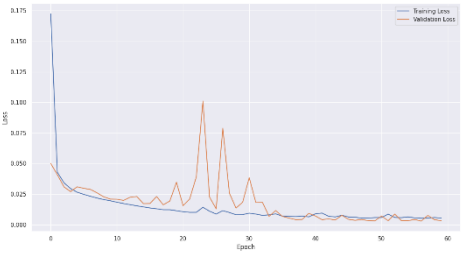


ภาพ 36 ช่วงข้อมูลของ Hyperbolic Tangent

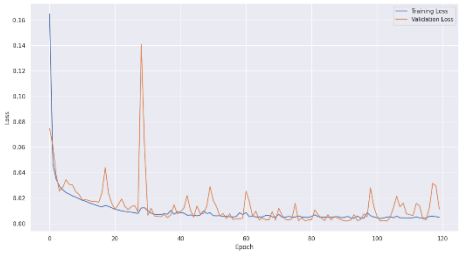
การกำหนดพามิเตอร์ภายนอกโครงสร้างของโมเดล LSTM แบบพื้นฐานนั้นจะประกอบไปด้วยสามขั้นตอนได้แก่ การทำ Dropout การหาจำนวนรอบของการฝึกฝนและการหาจำนวนของ Batch size โดยทั้งสามพารามิเตอร์สามารถอธิบายได้ดังนี้

1) การทำ Dropout [124] คือ เทคนิคหนึ่งในการป้องกันการเกิด Overfitting ในโมเดล Neural Network หรือการเรียนรู้เชิงลึก โดยการใช้ Dropout เป็นการสุ่มปิด (Drop) บางส่วนของโหนดหรือหน่วยย่อยต่างๆใน Hidden Layers ขณะที่กำลังฝึกโมเดล โดยการทดลองครั้งนี้ได้ทำการกำหนดค่าของการ Dropout 0.25 [125] หมายความว่าในแต่ละรอบของการฝึกโมเดล โมเดลจะสุ่มปิด (Drop) ร้อยละ25 ของบางส่วนของโหนดใน Hidden Layers ซึ่งอาจเป็นส่วนของการปิดของทั้ง Hidden Layers หรือบาง Hidden Layers ขึ้นอยู่กับว่าค่า Dropout ที่กำหนดในชั้นใดของโมเดล

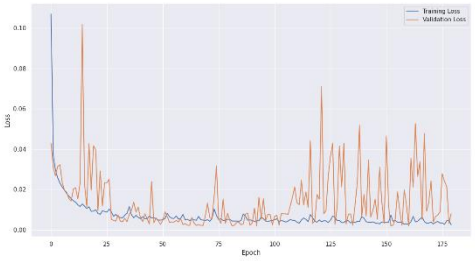
2) การกำหนดจำนวนรอบของการฝึกฝนของโมเดล (Epoch) นั้นไม่มีการกำหนดที่แน่ชัดและไม่ตายตัวเนื่องจากการพัฒนาโมเดลต่างๆจำเป็นต้องมีการทดลองการกำหนดจำนวนรอบที่เหมาะสมข้อมูลที่มีอยู่ซึ่งจะมีความสัมพันธ์กับการฝึกฝนของโมเดลโดยตรง ในแต่ละ Epoch โมเดลจะได้รับข้อมูลฝึกทั้งหมดและทำการอัปเดตพารามิเตอร์[114] เพื่อปรับให้โมเดลมีความสามารถในการแยกแยะหรือทำนายข้อมูลที่ดีขึ้น การฝึกโมเดลในแต่ละ Epoch จะเป็นการใช้ชุดข้อมูลฝึกที่ไม่เหมือนกันในแต่ละรอบ การฝึกในแต่ละ Epoch จะดำเนินการไปเรื่อยๆ จนกว่าจำนวน Epoch ที่กำหนดจะถึง และหลังจากนั้นจะสิ้นสุดการฝึกโมเดลหรือเรียกว่าเสร็จสิ้นการฝึกโดยที่โมเดลจะได้เรียนรู้ความสัมพันธ์ในข้อมูลที่มีความซับซ้อนมากขึ้นหลังจากที่ฝึกด้วยข้อมูลหลายรอบ ผู้วิจัยได้เปรียบเทียบการกำหนดค่า Epoch ดังภาพ 37



(ก)



(ข)



(ค)

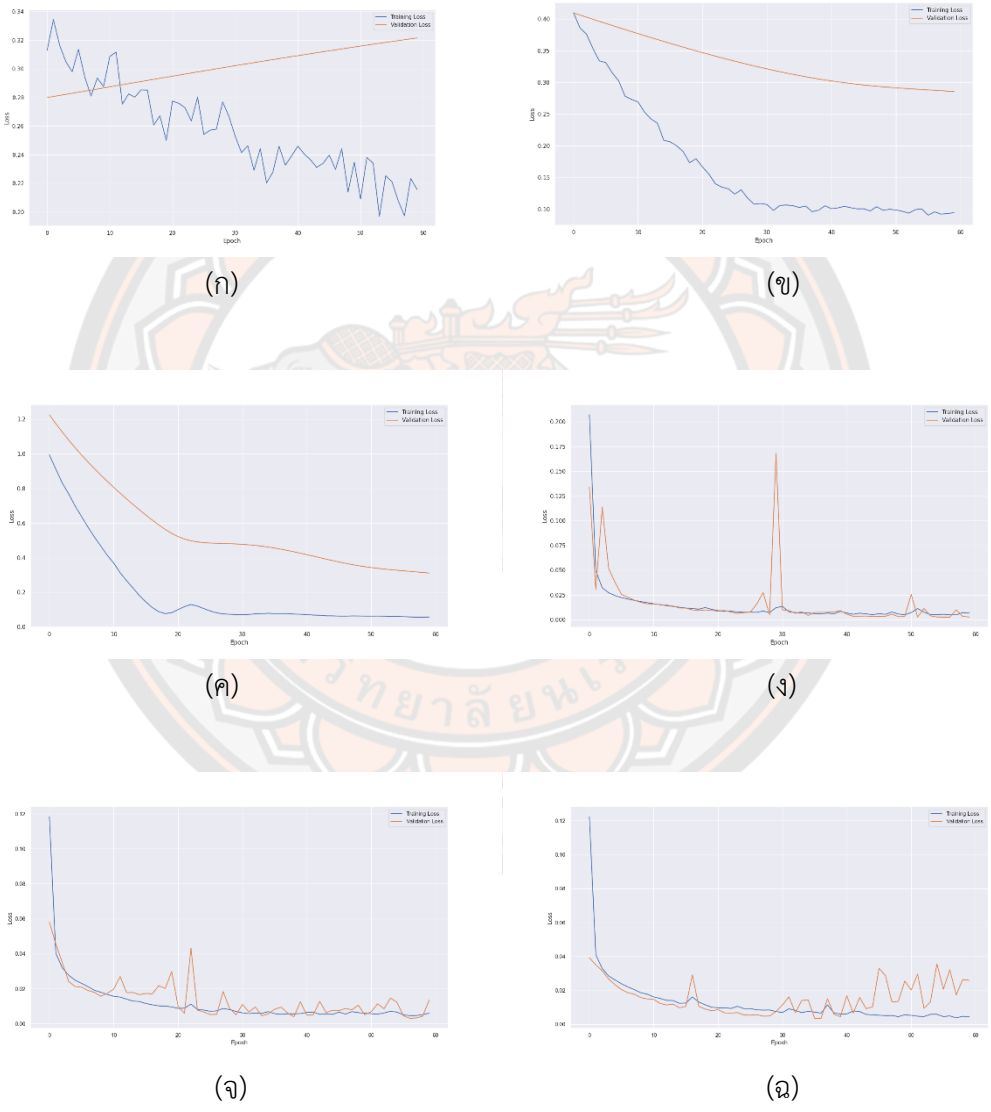
ภาพ 37 การกำหนดรอบการฝึกฝนข้อมูล
ภาพ ก คือ จำนวนรอบการฝึกที่ 60 ภาพ ข คือ จำนวนรอบการฝึกที่ 120 และ ภาพ ค คือ จำนวนรอบการฝึกที่ 180

จากการวิเคราะห์ Learning Curve ของโมเดล LSTM แบบพื้นฐานที่ทดสอบในจำนวน Epoch ต่าง ๆ ได้แก่ 60, 120, และ 180 Epoch ดังแสดงในภาพ 37 พบว่าทุกโมเดลมีการเกิด Overfitting ซึ่งเห็นได้ชัดเจนจากการที่เส้น Validation Loss เริ่มผันผวนและมีค่าเพิ่มสูงขึ้น ในขณะที่เส้น Training Loss ยังคงลดลงอย่างต่อเนื่องในช่วงท้ายของการฝึกฝน การที่เส้น Validation Loss สูงกว่า Training Loss ชี้ให้เห็นว่าโมเดลไม่สามารถเรียนรู้รายละเอียดของข้อมูลได้ ส่งผลให้ประสิทธิภาพในการทำนายข้อมูลใหม่ลดลง อย่างไรก็ตาม เมื่อพิจารณาถึงจำนวน Epoch ที่เหมาะสมในการฝึกฝน พบว่า 60 Epoch เพียงพอสำหรับการเรียนรู้ของโมเดล โดยแม้ว่าจะเกิด Overfitting จึงไม่จำเป็นต้องใช้จำนวน Epoch ที่มากขึ้น เนื่องจากการฝึกฝนโมเดลในจำนวน Epoch ที่มากกว่า 60 ไม่ได้ช่วยปรับปรุงผลลัพธ์ได้อย่างมีนัยสำคัญและยังใช้ทรัพยากรในการประมวลผลที่มากกว่า ดังนั้น การหยุดการฝึกฝนที่ 60 Epoch จึงเหมาะสมที่สุดในการลดความซับซ้อนและการใช้ทรัพยากรของโมเดล โดยหลังจากนี้จะมุ่งเน้นไปที่การปรับพารามิเตอร์อื่น ๆ ของ LSTM เพื่อเพิ่มประสิทธิภาพและลดการเกิด Overfitting การปรับแต่งพารามิเตอร์สำหรับโมเดล LSTM แบบพื้นฐานจึงกำหนดให้ใช้ 60 Epoch

3) การทำหนด Batch Size

การกำหนด Batch Size เป็นหนึ่งในพารามิเตอร์ที่สำคัญในการเรียนรู้ของแบบโมเดล [127] โดย Batch Size คือพารามิเตอร์ที่กำหนดจำนวนตัวอย่างของชุดข้อมูลสำหรับการฝึกฝนที่จะนำมาใช้ในการอัปเดตพารามิเตอร์ภายในของโมเดล โดยการกำหนดค่าของ Batch Size นั้นมีความสำคัญเป็นอย่างมากในการให้โมเดลมีประสิทธิภาพของโมและยังส่งผลไปถึงการใช้ทรัพยากรในการประมวลผล

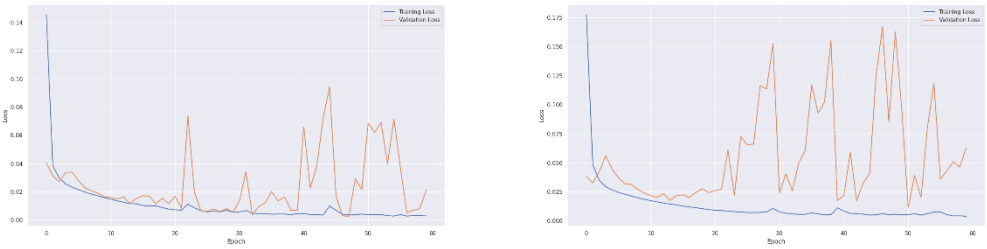
อีกด้วย โดยการกำหนด Batch Size นั้นจะเริ่มต้นจาก 128 ไปจนถึง 1280 เนื่องจากต้องสังเกตุดูการเปลี่ยนแปลงของ Learning Curve ของโมเดลว่า Batch Size ขนาดใดที่เหมาะสมกับข้อมูลที่ใช้ในการพัฒนาโมเดล โดยผลลัพธ์การเลือก Batch Size ที่ดีที่สุดสามารถเลือกได้จากการเปรียบเทียบดังแสดงในภาพ 38



(ก) (ข)

(ค) (ง)

(จ) (ฉ)



(ช) (ซ)

ภาพ 38 Learning Curve จากการกำหนด Batch Size ของโมเดล LSTM แบบพื้นฐาน ก) Batch Size ขนาด 128 ข) Batch Size ขนาด 256 ค) batch size ขนาด 400 ง) Batch Size ขนาด 640 จ) Batch Size ขนาด 720 ฉ) Batch Size ขนาด 800 ช) Batch Size ขนาด 960 ซ) Batch Size ขนาด 1280

จากภาพ 38 การวิเคราะห์กราฟที่แสดงผลของการฝึกโมเดลโดยใช้ Batch Size ที่แตกต่างกัน สามารถสังเกตเห็นว่าขนาด Batch Size ที่แตกต่างกันส่งผลต่อรูปแบบการลดลงของ Training Loss และ Validation Loss ในกราฟที่ใช้ Batch Size ต่ำกว่า 720 ภาพ จ เช่น Batch Size 128 ภาพ ก และ 256 ภาพ ข พบว่าเส้นของ Training Loss และ Validation Loss มีลักษณะที่แตกต่างกันอย่างชัดเจน โดยเส้นของ Training Loss มีการลดลงอย่างมาก ขณะที่เส้นของ Validation Loss ยังคงเพิ่มขึ้นหรือลดลงในอัตราที่ไม่เสถียร ซึ่งบ่งบอกถึงการที่โมเดลยังคงไม่สามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อนอย่างมีประสิทธิภาพและเกิดการ Overfitting

เมื่อใช้ Batch Size ที่มากกว่า 720 เช่น 800 ภาพ ฉ และ 960 ภาพ ช กลับพบว่าเส้นของ Validation Loss เริ่มแยกออกจากเส้นของ Training Loss อย่างชัดเจน ซึ่งหมายความว่าโมเดลอาจเริ่มมีปัญหาในการนำสิ่งที่เรียนรู้จากข้อมูลการฝึกไปประยุกต์ใช้กับข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน การแยกออกของเส้น Validation Loss เป็นสัญญาณบ่งบอกว่าโมเดลอาจเริ่มเกิดปัญหา Overfitting หรือไม่สามารถเรียนรู้จากข้อมูลการฝึกได้อย่างมีประสิทธิภาพเท่าที่ควร ดังนั้น Batch Size 720 จึงถือว่าเป็นขนาดที่เหมาะสมที่สุดในกรณีนี้ โดยสามารถลดปัญหาการผันผวนได้ดีกว่าการกำหนด Batch Size ค่าอื่นๆ

การใช้ Batch Size 720 ผลลัพธ์ที่ได้กลับมาพบว่าโมเดลสามารถปรับตัวได้ดีขึ้น เส้นของ Training Loss และ Validation Loss เริ่มมีความสอดคล้องกันมากขึ้น และไม่มีการผันผวนมากเกินไปในเส้นของ Validation Loss เมื่อเทียบกับค่าอื่นๆ ซึ่งถือเป็นสิ่งที่ดีว่าโมเดลเริ่มสามารถเรียนรู้จากข้อมูลการฝึกและนำไปใช้ทำนายข้อมูลใหม่ได้ แต่ยังเกิดปัญหา Overfitting เมื่อพิจารณาจาก

กราฟ Learning Curve การปรับแต่งพารามิเตอร์สำหรับโมเดล LSTM แบบพื้นฐานจึงกำหนดให้ใช้ Batch Size 720

จากการปรับค่าพารามิเตอร์เพื่อค่าที่ดีที่สุดในแต่ละค่าเพื่อให้สอดคล้องกับข้อมูลที่นำมาใช้ในการพัฒนาโมเดลเพื่อการทำนายในโมเดล LSTM แบบพื้นฐานเพื่อให้ผลลัพธ์ที่ดีที่สุด โดยผลลัพธ์การปรับแต่งพารามิเตอร์ได้ตามตาราง 5

ตาราง 5 การกำหนดค่าของโมเดลก่อนทำการประมวลผลกับชุดข้อมูล

โครงสร้าง LSTM	**ค่าที่กำหนด**
หน่วยย่อยภายใน LSTM เลเยอร์	
feature_range	(-1,1)
values.reshape	(-1,1)
model	Sequential
LSTM Unit	600
activation function	TanH
Regularizers l2	0.01
validation	0.1
optimizer	adam
loss function	MSE
หน่วยย่อยภายนอก LSTM เลเยอร์	
drop out	0.25
epoch	60
Batch Size	720

ตาราง 5 อธิบายการกำหนดพารามิเตอร์ของโมเดล LSTM แบบพื้นฐานที่ได้จากการปรับปรุงโมเดลโดยการปรับแต่งค่าพารามิเตอร์เพื่อให้เกิดผลลัพธ์ที่ดีที่สุดของโมเดล LSTM แบบพื้นฐานในโดยจะประเมินผลของโมเดลผ่านการวัดประสิทธิภาพของโมเดลผ่าน RMSE และ MAE และการวัดความเสถียรของโมเดลผล Learning Curve โดยค่าพารามิเตอร์ต่างๆอธิบายได้ดังนี้

จากการวิเคราะห์ทั้งหมด จึงได้ตัดสินใจเลือกโมเดล LSTM ที่มี 600 หน่วย เนื่องจากมีความเสถียรใน Validation Loss มากที่สุด แม้จะยังคงเกิดปัญหา Overfitting อยู่ แต่เป็นโมเดลที่สามารถรักษาความสมดุลระหว่างการเรียนรู้และการทำนายข้อมูลใหม่ได้ดีที่สุดในชุดโมเดลที่ทดสอบของ LSTM แบบพื้นฐาน อย่างไรก็ตาม เพื่อแก้ไขปัญหา Overfitting ที่ยังคงมีอยู่ การเพิ่ม LSTM เลเยอร์

อื่นเข้าไปในโมเดลอาจเป็นวิธีที่ดีในการปรับปรุงความสามารถในการทำนายของโมเดลให้ดีขึ้นและลดปัญหา Overfitting ที่เกิดขึ้น

จากกำหนดค่าพารามิเตอร์ดังตาราง 5 ที่แสดงการกำหนดค่าของโมเดล LSTM แบบพื้นฐานสำหรับการทำนายข้อมูล แม้ว่าจะมีการปรับค่าพารามิเตอร์ต่าง ๆ อย่างละเอียด แต่โมเดล LSTM แบบพื้นฐานยังคงมีข้อจำกัดในการทำนายข้อมูลที่มีความซับซ้อนสูง การปรับค่าพารามิเตอร์ เช่น ขนาดของหน่วย LSTM Activation Function Dropout จำนวน Epoch และ Batch Size อาจช่วยปรับปรุงประสิทธิภาพของโมเดลได้บ้าง แต่ก็ยังไม่เพียงพอที่จะทำให้โมเดลมีความแม่นยำที่ต้องการ หนึ่งในปัญหาหลักที่เกิดขึ้นคือ โมเดล LSTM แบบพื้นฐานไม่สามารถจดจำลำดับข้อมูลในระยะยาวได้ดีพอ ทำให้ผลลัพธ์การทำนายมีความคลาดเคลื่อนและอาจเกิดการ Overfitting ซึ่งหมายถึงการที่โมเดลจำค่าของข้อมูลฝึกมากเกินไป ทำให้ไม่สามารถทั่วไปกับข้อมูลใหม่ได้ดี การแก้ไขปัญหานี้สามารถทำได้โดยการเพิ่มชั้น LSTM เลเยอร์ในโมเดลและการใช้โมเดล MLP ที่สามารถจัดการข้อมูลแบบคงที่เพื่อเข้ามาช่วยในส่วนของความหลากหลายของข้อมูล

3.4.3 โมเดล LSTM + MLP

ในการพัฒนาโมเดล LSTM + MLP เป็นการพัฒนาและปรับปรุงจากโมเดล LSTM แบบพื้นฐาน เพื่อเพิ่มประสิทธิภาพในการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 โดยหนึ่งในปัญหาที่พบเมื่อใช้โมเดล LSTM แบบพื้นฐานคือการเกิด Overfitting เมื่อโมเดลไม่สามารถเรียนรู้ความซับซ้อนของข้อมูลได้ เพื่อแก้ไขปัญหานี้ การปรับปรุงโมเดลได้ทำโดยการเพิ่มจำนวนเลเยอร์ ของ LSTM ซึ่งช่วยในการรักษาความสม่ำเสมอของการเรียนรู้และลดความเสี่ยงของการเกิด Overfitting นอกจากนี้ ยังได้มีการผสานโมเดล MLP เข้ามาในโครงสร้างของโมเดลเพื่อจัดการข้อมูลแบบคงที่ การเพิ่ม MLP นี้ช่วยให้โมเดลมีความสามารถในการจัดการกับความหลากหลายของข้อมูลได้มากยิ่งขึ้น

การปรับปรุงโมเดล LSTM ในขั้นตอนเป็นการทดสอบการเพิ่มจำนวนชั้นของโมเดล LSTM ในการทำปรับปรุงโมเดลมีวัตถุประสงค์เพื่อสังเกตและประเมินการทำงานของโมเดลเมื่อมีการปรับเพิ่มชั้นมากขึ้นเรื่อยๆ การเพิ่มจำนวนชั้นของโมเดลสามารถช่วยให้โมเดลจับความซับซ้อนและลักษณะเฉพาะของข้อมูลได้ดีขึ้น โดยที่แต่ละชั้นเพิ่มเติมจะทำหน้าที่ดึงคุณลักษณะเชิงลึกจากข้อมูลออกมาได้มากขึ้น การทดลองเพิ่มชั้นในโมเดลเป็นการทดสอบเพื่อดูว่าโมเดลจะสามารถปรับปรุงประสิทธิภาพในการทำนายได้หรือไม่ และสามารถลดค่า Loss ของชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้องได้อย่างมีประสิทธิภาพหรือไม่

นอกจากนี้ยังเป็นการตรวจสอบว่าโมเดลจะเกิดปัญหา Overfitting หรือไม่ ซึ่งเป็นปัญหาที่โมเดลเรียนรู้รายละเอียดในชุดข้อมูลฝึกอบรมมากเกินไปจนไม่สามารถทำนายข้อมูลใหม่ได้ดี การเพิ่มจำนวนชั้นในโมเดล LSTM ช่วยให้สามารถสังเกตผลกระทบต่อค่า Loss และประสิทธิภาพของโมเดล

เมื่อมีการเพิ่มชั้นมากขึ้น ทำให้สามารถปรับแต่งและหาโครงสร้างที่เหมาะสมที่สุดสำหรับการทำนายข้อมูลที่มีความซับซ้อนได้อย่างมีประสิทธิภาพโดยในการทดลองนั้นจะใช้ชุดข้อมูลของการติดเชื้อสะสมเพื่อเป็นการทำลองหากการเพิ่มจำนวนชั้นไม่สามารถทำงานได้ดีกับชุดข้อมูลการติดเชื้อสะสมได้นั้นจะไม่ได้นำไปใช้กับชุดข้อมูลการเสียชีวิตสะสมเนื่องจากความซับซ้อนของชุดข้อมูลการติดเชื้อสะสมนั้นไม่ซับซ้อนเท่ากับชุดข้อมูลการติดเชื้อสะสม ผลการทดลองการเพิ่มเลเยอร์ของโมเดล LSTM สามารถอธิบายได้ดังนี้

1) LSTM แบบ 1 เลเยอร์

การทดลอง LSTM แบบ 1 เลเยอร์ คือการสังเกตุ Learning Curve ของโมเดล LSTM โดยมี LSTM แบบ 1 เลเยอร์ เพื่อวิเคราะห์การลดลงของค่า Loss ดังแสดงในภาพ 39

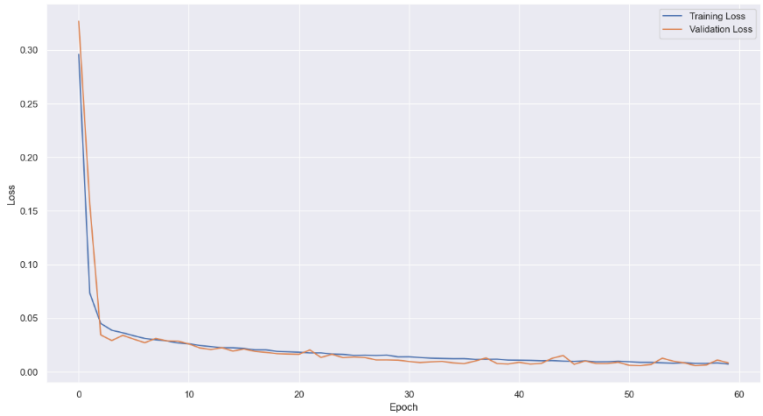
ภาพ 39 Learning Curve ของโมเดล LSTM แบบ 1 เลเยอร์

ภาพ 39 เป็นการเปรียบเทียบค่า Loss ระหว่างชุดข้อมูลฝึกอบรม (Training Loss) และชุดข้อมูลตรวจสอบความถูกต้อง (Validation Loss) ตลอดระยะเวลา 60 Epoch ของการฝึกโมเดล LSTM แบบ 1 เลเยอร์ เพื่อทำนายจำนวนผู้ติดเชื้อสะสม แกน X ของกราฟนี้แสดงจำนวน Epoch ซึ่งหมายถึงจำนวนครั้งที่โมเดลได้ทำการเรียนรู้หรือฝึกฝนจากชุดข้อมูลทั้งหมด ส่วนแกน Y แสดงค่าความสูญเสีย (Loss) ซึ่งเป็นการวัดความคลาดเคลื่อนระหว่างค่าที่โมเดลทำนายกับค่าจริง จากกราฟจะเห็นได้ว่าค่า Training Loss ลดลงอย่างรวดเร็วในช่วงแรกของการฝึกฝน และค่อยๆ ลดลงอย่างช้าๆ เมื่อจำนวน Epoch เพิ่มขึ้น แสดงให้เห็นว่าโมเดลมีการเรียนรู้และปรับปรุงผลการทำนายให้แม่นยำมากขึ้นตามเวลา อย่างไรก็ตาม เมื่อพิจารณาค่า Validation Loss จะสังเกตเห็นว่ามีแนวโน้มที่จะลดลงในช่วงแรกเช่นกัน แต่หลังจากนั้นค่า Validation Loss จะเริ่มมีความผันผวนมากขึ้นและ

ไม่ได้ลดลงอย่างสม่ำเสมอ แนวโน้มนี้บ่งชี้ถึงการเกิด Overfitting ในโมเดล ซึ่งหมายความว่าโมเดลเรียนรู้รายละเอียดในชุดข้อมูลฝึกอบรมมากเกินไป ทำให้โมเดลมีความแม่นยำสูงเมื่อทำนายชุดข้อมูลฝึกอบรม แต่เมื่อทำนายชุดข้อมูลใหม่หรือชุดข้อมูลตรวจสอบความถูกต้อง ค่า Loss จะไม่สม่ำเสมอและอาจเพิ่มขึ้นหรือผันผวนมากขึ้น สรุปได้ว่า โมเดล LSTM นี้เกิดการ Overfitting เนื่องจากค่า Validation Loss มีความผันผวนมากขึ้นในขณะที่ค่า Training Loss ลดลงอย่างต่อเนื่อง

2) LSTM แบบ 2 เลเยอร์

การทดลอง LSTM แบบ 2 เลเยอร์ คือการสังเกตุ Learning Curve ของโมเดล LSTM โดยมี LSTM แบบ 2 เลเยอร์ เพื่อวิเคราะห์การลดลงของค่า Loss ดังแสดงในภาพ 40



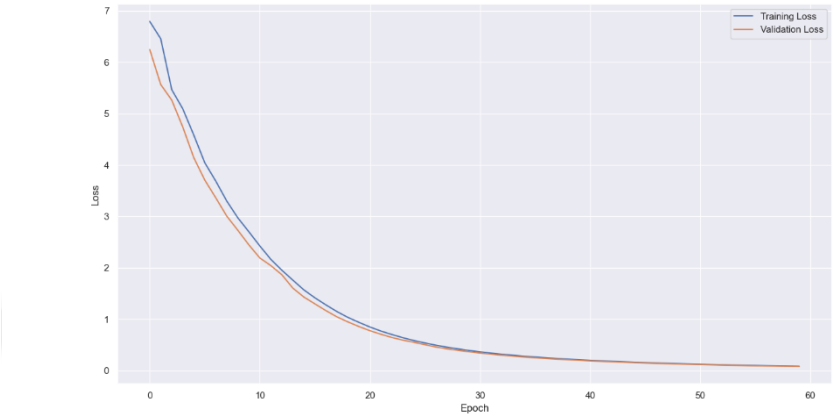
ภาพ 40 Learning Curve ของโมเดล LSTM แบบ 2 เลเยอร์

กราฟที่แสดงในภาพ 40 เป็นการเปรียบเทียบค่า Loss ระหว่างชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้อง ตลอดระยะเวลา 60 Epoch ของการฝึกโมเดล LSTM แบบ 2 เลเยอร์ เพื่อทำนายจำนวนผู้ติดเชื้อสะสม แกน X ของกราฟนี้แสดงจำนวน Epoch ซึ่งหมายถึงจำนวนครั้งที่โมเดลได้ทำการเรียนรู้หรือฝึกฝนจากชุดข้อมูลทั้งหมด ส่วนแกน Y แสดงค่าความสูญเสีย (Loss) ซึ่งเป็นการวัดความคลาดเคลื่อนระหว่างค่าที่โมเดลทำนายกับค่าจริง จากกราฟจะเห็นได้ว่าค่า Training Loss และ Validation Loss ลดลงอย่างรวดเร็วในช่วงแรกของการฝึกฝน และค่อยๆ ลดลงอย่างช้าๆ เมื่อจำนวน Epoch เพิ่มขึ้น แสดงให้เห็นว่าโมเดลมีการเรียนรู้และปรับปรุงผลการทำนายให้แม่นยำมากขึ้นตามเวลา แนวโน้มนี้แสดงให้เห็นว่าโมเดลไม่ได้เกิดการ Overfitting เนื่องจากค่า Training Loss และ Validation Loss มีแนวโน้มใกล้เคียงกันตลอดระยะเวลาการฝึกฝน และค่า Validation

Loss ไม่ได้ผันผวนมากนัก โมเดลนี้สามารถทำนายผลได้ดีทั้งในชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้อง โดยไม่ได้มีการเรียนรู้รายละเอียดและสัญญาณรบกวน (Noise) ในชุดข้อมูลฝึกอบรมมากเกินไป

3) LSTM แบบ 3 เลเยอร์

การทดลอง LSTM แบบ 3 เลเยอร์ เป็นการสังเกตุ Learning Curve ของโมเดล LSTM โดยมี LSTM แบบ 3 เลเยอร์ เพื่อวิเคราะห์การลดลงของค่า Loss ดังแสดงในภาพ 41

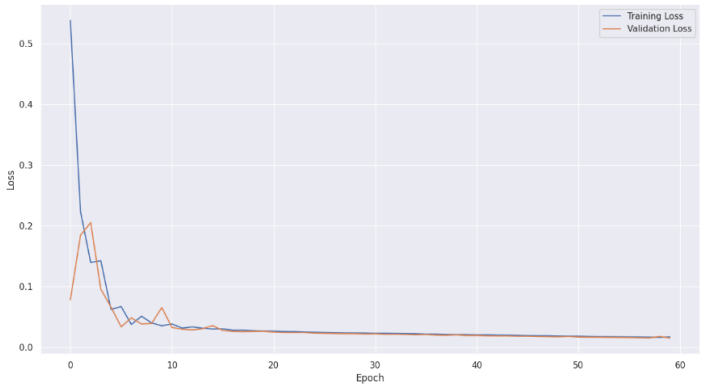


ภาพ 41 Learning Curve ของโมเดล LSTM แบบ 3 เลเยอร์

ภาพ 41 แสดงการเปรียบเทียบค่า Loss ระหว่างชุดข้อมูลฝึกอบรม (Training Loss) และชุดข้อมูลตรวจสอบความถูกต้อง (Validation Loss) ตลอดระยะเวลา 60 Epoch ของการฝึกโมเดล LSTM แบบ 3 เลเยอร์ เพื่อทำนายจำนวนผู้ติดเชื้อสะสม แกน X แสดงจำนวน Epoch ซึ่งหมายถึงจำนวนครั้งที่โมเดลได้ทำการเรียนรู้หรือฝึกฝนจากชุดข้อมูลทั้งหมด ส่วนแกน Y แสดงค่าความสูญเสีย (Loss) ซึ่งเป็นการวัดความคลาดเคลื่อนระหว่างค่าที่โมเดลทำนายกับค่าจริง จากกราฟจะเห็นได้ว่าค่า Training Loss และ Validation Loss ลดลงอย่างสม่ำเสมอและไม่มีความผันผวนมากนัก แสดงถึงความเสถียรของโมเดลนี้ ค่า Validation Loss ลดลงอย่างต่อเนื่องและเกือบขนานกับค่า Training Loss ซึ่งบ่งบอกถึงความแม่นยำในการทำนายและไม่มีการเกิด Overfitting

4) LSTM แบบ 4 เลเยอร์

การทดลอง LSTM แบบ 4 เลเยอร์ คือการสังเกตุ Learning Curve ของโมเดล LSTM โดยมี LSTM แบบ 4 เลเยอร์ เพื่อวิเคราะห์การลดลงของค่า Loss ดังแสดงในภาพ 42

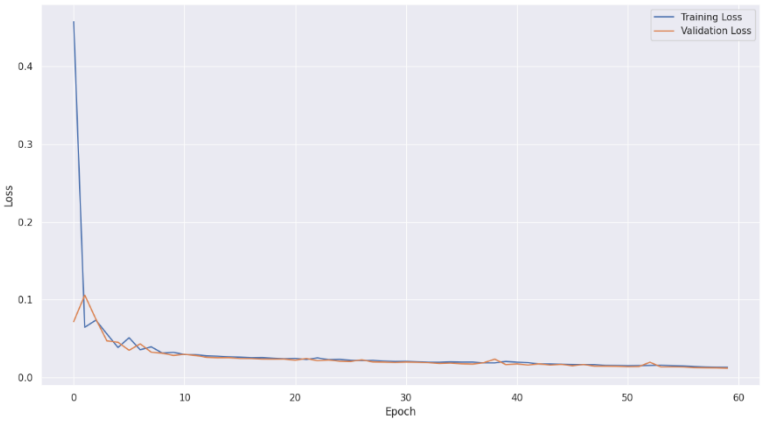


ภาพ 42 Learning Curve ของโมเดล LSTM แบบ 4 เลเยอร์

การเปรียบเทียบค่า Loss ดังแสดงในภาพ 42 ระหว่างชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้อง ตลอดระยะเวลา 60 Epoch ของการฝึกโมเดล LSTM แบบ 4 เลเยอร์ เพื่อทำนายจำนวนผู้ติดเชื้อสะสม จากกราฟจะเห็นได้ว่าค่า Training Loss และ Validation Loss ลดลงอย่างรวดเร็วในช่วงแรกของการฝึกฝน โดยช่วง Epoch ที่ 0 ถึง 10 มีความผันผวนอยู่บ้าง หลังจากนั้นค่า Validation Loss จะค่อนข้างคงที่และมีความผันผวนเล็กน้อยในบางช่วง ซึ่งเป็นช่วง Epoch ที่ 10 ถึง 20 แนวโน้มนี้แสดงให้เห็นว่าโมเดล LSTM แบบ 4 เลเยอร์ไม่ได้เกิดการ Overfitting เนื่องจากค่า Training Loss และ Validation Loss มีแนวโน้มใกล้เคียงกันตลอดระยะเวลาการฝึกฝน และค่า Validation Loss ไม่ได้ผันผวนมากนัก โมเดลนี้สามารถทำนายผลได้ดีทั้งในชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้อง ทำให้มั่นใจได้ว่าโมเดลนี้มีประสิทธิภาพและความเสถียรในการทำนายผู้ติดเชื้อสะสม

5) LSTM แบบ 5 เลเยอร์

การทดลอง LSTM แบบ 5 เลเยอร์ คือการสังเกตุ Learning Curve ของโมเดล LSTM โดยมี LSTM แบบ 5 เลเยอร์ เพื่อวิเคราะห์การลดลงของค่า Loss ดังแสดงในภาพ 43



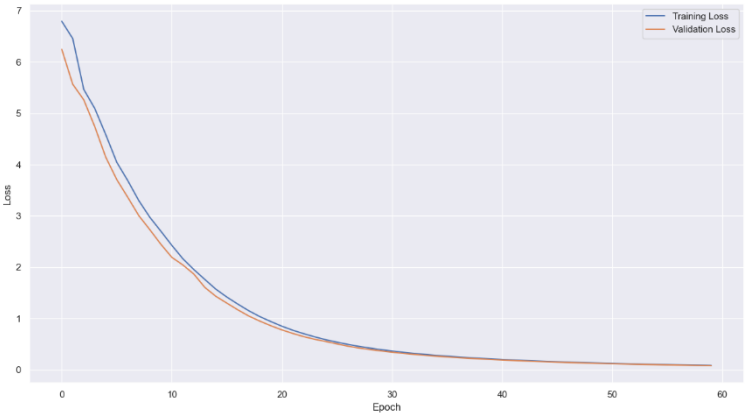
ภาพ 43 Learning Curve ของโมเดล LSTM แบบ 5 เลเยอร์

จากกราฟที่แสดงการเปรียบเทียบค่า Loss ระหว่างชุดข้อมูลฝึกอบรมและชุดข้อมูลตรวจสอบความถูกต้อง ของโมเดล LSTM แบบ 4 เลเยอร์และ 5 เลเยอร์ พบว่ามีช่วงที่ไม่เสถียรในทั้งสองกราฟ ดังนี้ สำหรับโมเดล LSTM แบบ 4 เลเยอร์ จะพบว่าความไม่เสถียรปรากฏอยู่ในช่วง Epoch ที่ 0 ถึง 10 โดยค่า Validation Loss มีความผันผวนอยู่บ้างหลังจากนั้นค่า Validation Loss จะลดลงและค่อนข้างคงที่ในช่วง Epoch ที่เหลือ ทำให้โมเดลนี้มีความเสถียรในการทำนายมากขึ้นเมื่อผ่านช่วง Epoch แรกไปแล้ว ในขณะที่โมเดล LSTM แบบ 5 เลเยอร์ พบว่ามีช่วงความไม่เสถียรในช่วง Epoch ที่ 0 ถึง 10 เช่นกัน โดยค่า Validation Loss มีความผันผวนในระดับที่ใกล้เคียงกับโมเดล 4 เลเยอร์ แต่หลังจาก Epoch ที่ 10 ไปแล้ว ค่า Validation Loss ของโมเดล 5 เลเยอร์ยังคงมีความผันผวนเล็กน้อยในบางช่วง เมื่อเปรียบเทียบความเสถียรของทั้งสองโมเดล พบว่าโมเดล 4 เลเยอร์มีความเสถียรมากกว่า เนื่องจากค่า Validation Loss ของโมเดล 4 เลเยอร์มีความคงที่มากขึ้นหลังจากผ่านช่วง Epoch แรกไปแล้ว ในขณะที่โมเดล 5 เลเยอร์ยังคงมีความผันผวนอยู่บ้างในบางช่วงหลังจาก Epoch ที่ 10 ซึ่งแสดงให้เห็นว่าโมเดล 4 เลเยอร์สามารถจัดการกับความซับซ้อนของข้อมูลได้ดีและมีความแม่นยำสูงในการทำนายผล โดยไม่เกิดการผันผวนของค่า Validation Loss มากนัก

การเพิ่มชั้น LSTM เลเยอร์ช่วยให้โมเดลมีความสามารถในการประมวลผลและจดจำข้อมูลในระยะยาวได้ดีขึ้น เนื่องจากแต่ละชั้น LSTM จะช่วยในการเรียนรู้จากลำดับข้อมูลที่แตกต่างกัน ซึ่งทำ

ให้โมเดลสามารถจับความสัมพันธ์ในข้อมูลที่มีความซับซ้อนสูงได้ดีขึ้น นอกจากนี้ การเพิ่มชั้น LSTM เลเยอร์ยังช่วยลดปัญหาการเกิด Overfitting ทำให้โมเดลมีความสามารถในการทั่วไปกับข้อมูลใหม่ได้ดีขึ้น ส่งผลให้ผลลัพธ์การทำนายมีความแม่นยำมากขึ้น ดังนั้น การเพิ่มชั้น LSTM เลเยอร์เป็นวิธีที่มีประสิทธิภาพในการปรับปรุงความแม่นยำในการทำนายของโมเดล LSTM และลดการเกิดปัญหา Overfitting ซึ่งเป็นปัญหาที่เกิดขึ้นเมื่อใช้โมเดล LSTM แบบพื้นฐาน จากการทดลองการเปรียบการเพิ่มเลเยอร์ของโมเดล LSTM พบว่าโมเดล LSTM แบบ 3 เลเยอร์ให้ประสิทธิภาพมากที่สุดดังนั้น ผู้วิจัยจึงได้พัฒนาโมเดล LSTM + MLP โดยใช้ LSTM แบบ 3 เลเยอร์

การกำหนดจำนวนรอบการฝึก (Epoch) ของโมเดล LSTM หลังจากที่ได้พัฒนาโมเดลในการลดการเกิด Overfitting ของโมเดล LSTM มีรายละเอียดดังนี้



ภาพ 44 จำนวนรอบการฝึกฝนโมเดล LSTM ที่กำหนด

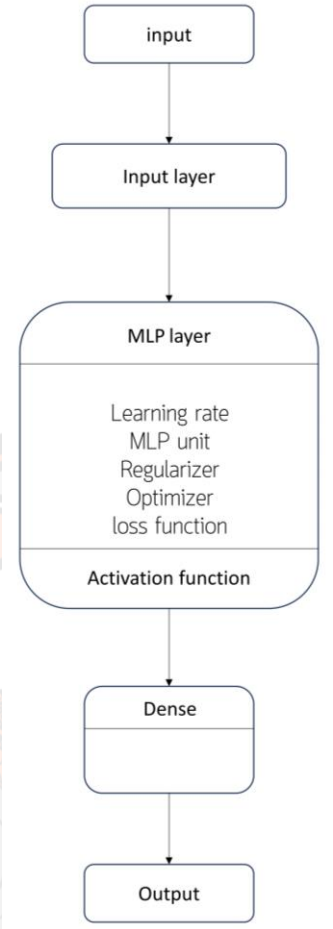
จากภาพ 44 คือแสดงการเปรียบเทียบค่า Loss ของทั้ง Training และ Validation ในแต่ละ Epoch สามารถวิเคราะห์ความเหมาะสมของการเลือก Epoch ที่ดีที่สุด ในช่วงแรกของการฝึกโมเดล (ประมาณ 10-30 Epoch) จะเห็นว่าค่า Loss ของทั้ง Training และ Validation ลดลงอย่างรวดเร็ว ซึ่งบ่งบอกว่าโมเดลกำลังเรียนรู้รูปแบบจากข้อมูลได้อย่างมีประสิทธิภาพ ในช่วงนี้ ค่า Training Loss และ Validation Loss ยังมีความแตกต่างกันพอสมควร โดยค่า Training Loss จะต่ำกว่าค่า Validation Loss ซึ่งเป็นเรื่องปกติเนื่องจากโมเดลกำลังพยายามปรับพารามิเตอร์ให้เหมาะสมกับข้อมูลที่มีอยู่ หลังจาก Epoch ที่ 30 เป็นต้นไป การลดลงของค่า Loss ทั้งสองเริ่มช้าลง และค่า Loss ของทั้ง Training และ Validation เริ่มใกล้เคียงกันมากขึ้น ซึ่งบ่งบอกว่าโมเดลเริ่มมีการ Convergence หรือการเข้าสู่สภาวะที่โมเดลมีการเรียนรู้ได้ใกล้เคียงกับศักยภาพสูงสุดของแล้ว ความ

ใกล้เคียงกันของค่า Loss ทั้งสองยังเป็นสิ่งที่บ่งบอกว่าโมเดลไม่เกิดการ Overfitting กับข้อมูล Training มากเกินไป เพราะค่า Validation Loss ไม่ได้เพิ่มขึ้นแต่อย่างใด

ในขณะที่ Epoch ที่ 60 ซึ่งเป็น Epoch สุดท้ายที่แสดงในกราฟ ค่า Loss ทั้งสองมีค่าต่ำและคงที่มากที่สุด โดยที่ค่า Validation Loss ต่ำที่สุดที่จุดนี้ นั่นหมายความว่า โมเดลในจุดนี้สามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อน (Validation Data) ได้ดีที่สุด ซึ่งทำให้ Epoch ที่ 60 เป็นตัวเลือกที่เหมาะสมที่สุดในการหยุดการฝึกโมเดล การเทรนโมเดลต่อไปหลังจากนี้อาจจะไม่ก่อให้เกิดการปรับปรุงประสิทธิภาพเพิ่มเติม และอาจเสี่ยงต่อการ Overfitting ได้ ดังนั้นการเลือกหยุดการฝึกโมเดลที่ Epoch ที่ 60 เป็นทางเลือกที่เหมาะสม เนื่องจากโมเดลได้ Convergence อย่างดีและมีประสิทธิภาพสูงสุดในการทำนายข้อมูล Validation ซึ่งเป็นเป้าหมายหลักในการเทรนโมเดล. ผู้วิจัยจึงได้กำหนดให้พารามิเตอร์ Epoch เป็น 60

จากการทดลองการเพิ่มจำนวนชั้นของโมเดล LSTM แล้วนั้นพบว่าโมเดล LSTM แบบ 3 เลเยอร์มีความแม่นยำที่สุด เมื่อได้ผลลัพธ์โมเดล LSTM ที่ดีที่สุดแล้วจะนำไปออกแบบการทำงานร่วมกับกับโมเดล MLP ที่มีหน้าที่จัดการข้อมูลในรูปแบบของคงที่เพื่อให้รองรับปัจจัยอื่นๆสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 ที่ครอบคลุมมากยิ่งขึ้น และกำหนดให้จำนวนรอบการฝึกฝนเป็น 60 Epoch โดยการทำโมเดล LSTM แบบ 3 เลเยอร์นั้นมาทดสอบทำงานร่วมกับโมเดล MLP เพื่อให้ทั้งสองโมเดลสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพในหัวข้อถัดไปเพื่อกำหนดโครงสร้างของโมเดล LSTM ที่ทำงานร่วมกับโมเดล MLP

การกำหนดโครงสร้างโมเดล MLP โดยจะเป็นโมเดลที่จะนำไปทำงานร่วมกับโมเดล LSTM ที่ได้ทำการปรับปรุงโมเดลในขั้นตอนก่อนหน้า โดยภายในโครงสร้างของ MLP เลเยอร์จะประกอบไปด้วยพารามิเตอร์ดังนี้ 1) หน่วยของ MLP 2) Regularizer 3) Optimizer 4) Loss function และ 5) Activation function พารามิเตอร์เหล่านี้ส่งผลต่อประสิทธิภาพการทำนายของโมเดล LSTM แบบพื้นฐาน จึงต้องมีการปรับแต่งเพื่อหาค่าพารามิเตอร์ที่ดีที่สุดของโมเดล โครงสร้างโมเดล MLP แสดงในภาพ 45



ภาพ 45 โครงสร้างของโมเดล MLP

การปรับปรุงโมเดล MLP เพื่อให้สามารถจัดการกับข้อมูลแบบคงที่ได้สำหรับการนำไปทำงานร่วมกับโมเดล LSTM การปรับปรุงโมเดล MLP นั้นเป็นขั้นตอนของการปรับค่าพารามิเตอร์ภายในโมเดล MLP ดังนี้ การกำหนดLearning Rate การกำหนด Regularizers การกำหนด Optimizer การกำหนด Batch Size การกำหนดหน่วยของ MLP และ การกำหนด MLP เลเยอร์ โดยจะแสดงขั้นตอนการกำหนดดังต่อไปนี้

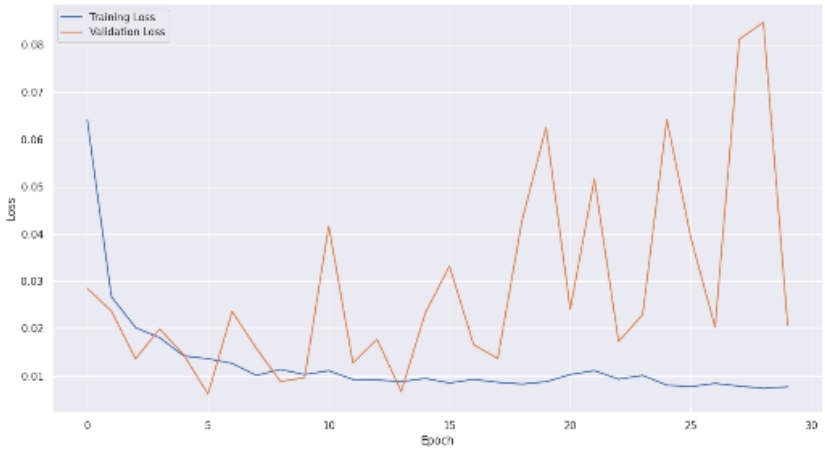
1) Learning Rate

การเลือกใช้ Learning Rate ที่ค่า 0.001 ในโมเดล MLP มีความสำคัญต่อกระบวนการฝึกโมเดล เนื่องจาก Learning Rate เป็นพารามิเตอร์ที่กำหนดขนาดของ Step Size ในแต่ละขั้นตอนของการปรับค่าพารามิเตอร์โมเดล เมื่อมีการคำนวณค่าความผิดพลาด (Loss) ในแต่ละรอบ การตั้งค่า Learning Rate ที่ 0.001 [128] เป็นการกำหนดให้โมเดลปรับค่าพารามิเตอร์ในอัตราที่ไม่เร็วหรือช้า

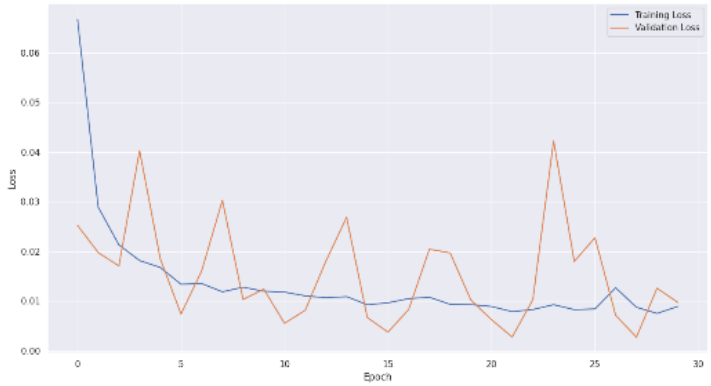
เกินไป ทำให้การเรียนรู้ของโมเดลมีความสมดุล ไม่เกิดการเปลี่ยนแปลงที่ใหญ่เกินไปในแต่ละรอบ ซึ่งอาจทำให้โมเดลข้ามค่าเหมาะสมที่สุด (Optimal Point) หรือไม่เกิดการเรียนรู้ที่ช้าเกินไปซึ่งอาจทำให้การฝึกโมเดลใช้เวลานานมากเกินไป การกำหนด Learning Rate ดังกล่าวช่วยให้โมเดลสามารถเรียนรู้จากข้อมูลได้อย่างมีประสิทธิภาพโดยลดความเสี่ยงของการติดอยู่ใน Local Minima และช่วยให้โมเดลมีความสามารถในการค้นหาค่าที่เหมาะสมที่สุดของพารามิเตอร์ต่าง ๆ ได้ดียิ่งขึ้น ค่า Learning Rate นี้เป็นค่าที่ได้รับความนิยมในการฝึกโมเดล MLP เนื่องจากมีการทดสอบและพบว่ามีประสิทธิภาพในการช่วยให้โมเดลเรียนรู้ได้ดีในหลากหลายกรณี ทั้งในแง่ของการลดค่า Loss อย่างต่อเนื่องและการเพิ่มความแม่นยำในการทำนายผล ผู้วิจัยจึงได้กำหนด Learning Rate ของโมเดล MLP ที่ 0.001

2) หน่วยของโมเดล MLP

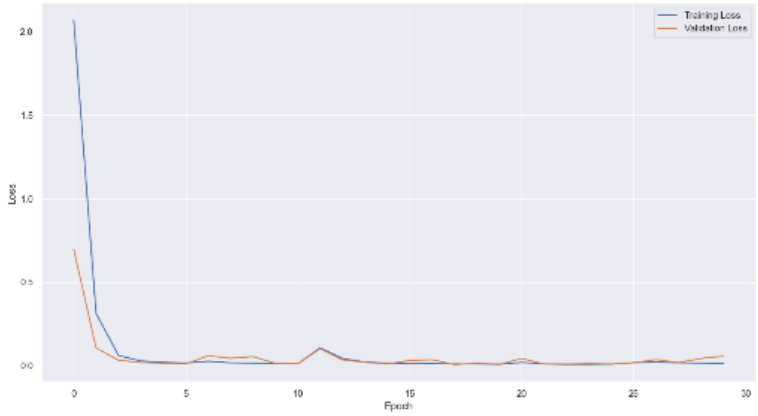
การกำหนดหน่วยของโมเดล MLP นั้นเป็นขั้นตอนที่สำคัญเนื่องจากจำเป็นต้องมีการสุ่มปรับค่าของหน่วยให้พอดีกับการทำงานของโมเดลและจำเป็นต้องทำงานสอดคล้องกับข้อมูลที่ใช้ฝึกโมเดล โดยโมเดล MLP นั้นจะทำการรับข้อมูลแบบคงที่เข้ามาทำการฝึกโมเดลแต่เนื่องจากข้อมูลแบบคงที่มีความซับซ้อนน้อยกว่าโมเดล LSTM ดังนั้นจึงไม่จำเป็นที่จะต้องใช้หน่วยที่สูงเท่ากับโมเดล LSTM โดยการทดลองการปรับหน่วยของโมเดล MLP แสดงในภาพ 46-48



ภาพ 46 กำหนด MLP ที่ 104 หน่วย



ภาพ 47 กำหนด MLP ที่ 112 หน่วย



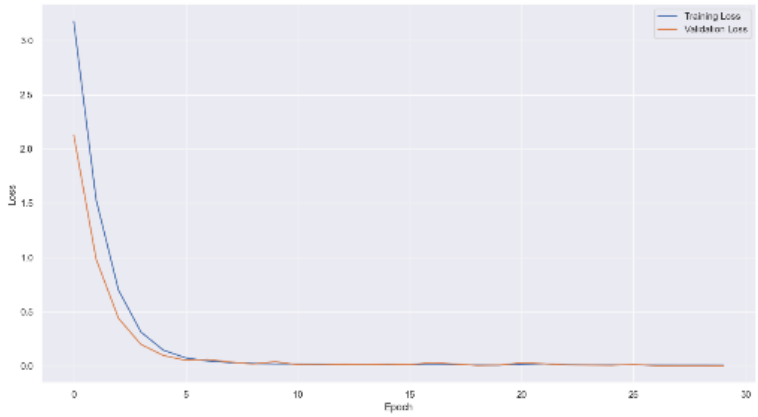
ภาพ 48 กำหนด MLP ที่ 120 หน่วย

จากการทดสอบการเพิ่มจำนวนของ MLP หน่วย โดยเริ่มจาก 104 หน่วยดังภาพ 46 พบว่าโมเดลมีความสามารถในการลดค่า Training Loss ได้อย่างรวดเร็วในช่วงแรกและคงที่ตลอดการฝึกสอน อย่างไรก็ตาม ค่า Validation Loss กลับแสดงความผันผวนสูงตลอดช่วงการเรียนรู้ ซึ่งบ่งบอกถึงความไม่เสถียรของโมเดลในการประเมินข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน ความผันผวนนี้เป็นสัญญาณว่าโมเดลอาจยังไม่ทั่วไปเพียงพอ และมีความเสี่ยงต่อการ Overfitting เมื่อใช้จำนวนหน่วยนี้

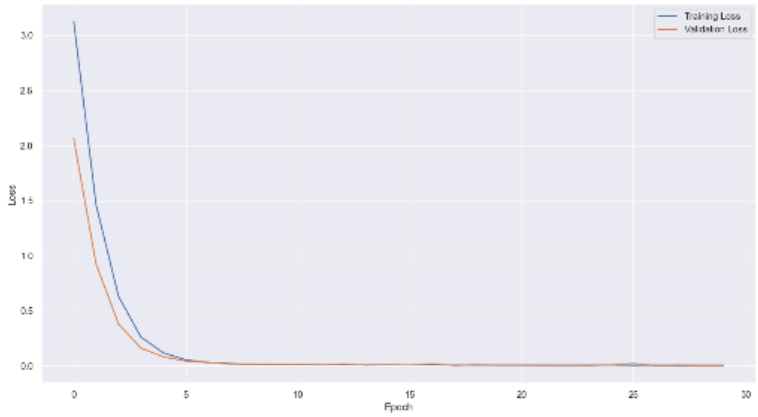
ภาพ 47 เมื่อเพิ่มจำนวนหน่วยเป็น 112 โมเดลแสดงผลการเรียนรู้ที่ดีขึ้นเล็กน้อย ค่า Validation Loss ลดความผันผวนลงเมื่อเทียบกับกรณีของ 104 หน่วยการลดลงของความผันผวนนี้บ่งบอกว่าโมเดลมีการเรียนรู้ที่เสถียรมากขึ้น และสามารถจัดการกับข้อมูล Validation ได้ดีขึ้น

อย่างไรก็ตาม ความผันผวนยังคงมีอยู่ในบางช่วง แสดงให้เห็นว่าการเพิ่มหน่วยในระดับนี้ช่วยปรับปรุงเสถียรภาพของโมเดล แต่ยังไม่สามารถขจัดปัญหาความผันผวนได้ทั้งหมด

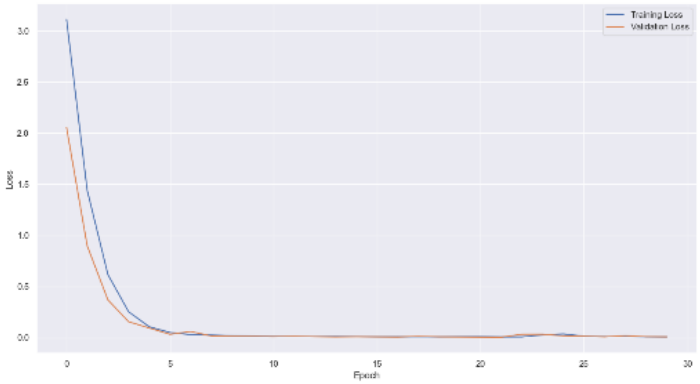
เมื่อเพิ่มจำนวนหน่วยเป็น 120 ดังภาพ 48 ผลลัพธ์แสดงถึงความเสถียรของโมเดลที่ดีขึ้นอย่างชัดเจน ค่า Validation Loss มีความผันผวนน้อยลงมาก และกราฟแสดงถึงความคงที่ของทั้ง Training Loss และ Validation Loss ตลอดการฝึกสอน



ภาพ 49 กำหนด MLP ที่ 128 หน่วย



ภาพ 50 กำหนด MLP ที่ 136 หน่วย



ภาพ 51 กำหนด MLP ที่ 144 หน่วย

จากการทดสอบและเปรียบเทียบผลลัพธ์การเพิ่มจำนวนหน่วยของ MLP โดยใช้จำนวนหน่วยที่แตกต่างกันระหว่าง 120 128 136 และ 144 พบว่ามีความเสถียรที่ใกล้เคียงกันในแง่ของการลดค่า Training Loss และ Validation Loss โดยแต่ละกรณีสามารถอธิบายได้ดังนี้

ภาพ 48 การใช้ 120 หน่วยพบว่าค่า Training Loss ลดลงอย่างรวดเร็วและคงที่หลังจากไม่กี่ Epoch ในขณะที่ค่า Validation Loss ก็ลดลงในลักษณะเดียวกัน แต่ยังมีความผันผวนเล็กน้อยในบางช่วง แสดงให้เห็นว่าโมเดลสามารถเรียนรู้ได้ดีแต่ยังคงมีความไม่เสถียรอยู่บ้างเมื่อประเมินข้อมูลใหม่ การใช้ 120 หน่วยนั้นเพียงพอสำหรับการทำให้โมเดลเรียนรู้ได้อย่างมีประสิทธิภาพ แต่ยังไม่สามารถลดความผันผวนของ Validation Loss ให้หมดไปได้อย่างสมบูรณ์

เมื่อเพิ่มจำนวนหน่วยเป็น 128 ผลลัพธ์ที่ได้แสดงถึงการปรับปรุงในด้านความเสถียรของโมเดลในภาพ 49 ค่า Validation Loss ลดลงอย่างมีนัยสำคัญและมีความคงที่มากขึ้น ทำให้โมเดลสามารถทำนายได้อย่างมีความแม่นยำมากขึ้น การลดลงของความผันผวนใน Validation Loss บ่งชี้ว่าโมเดลมีความเสถียรมากขึ้น และสามารถจัดการกับข้อมูลที่ไม่เคยเห็นมาก่อนได้ดีกว่าการใช้ 120 หน่วย

การเพิ่มหน่วยเป็น 136 ในภาพ 50 และ 144 ในภาพ 51 พบว่าค่า Training Loss และ Validation Loss ยังคงลดลงอย่างสม่ำเสมอและมีความเสถียรในระดับใกล้เคียงกันกับ 128 หน่วยไม่มีการเปลี่ยนแปลงที่ชัดเจนหรือมีนัยสำคัญเมื่อเพิ่มหน่วยจาก 128 เป็น 136 หรือ 144 แสดงว่าการเพิ่มจำนวนหน่วยในระดับนี้ไม่ได้ช่วยเพิ่มประสิทธิภาพของโมเดลอย่างชัดเจน การใช้ 128 หน่วยจะเป็นจุดที่ให้ผลลัพธ์ที่ดีที่สุดในแง่ของการรักษาสมดุลระหว่างความเสถียรของโมเดลและการใช้ทรัพยากรในการประมวลผล การเพิ่มจำนวนหน่วยทีละ 8 หน่วยในแต่ละครั้งช่วยให้การปรับปรุง

โมเดลทำได้อย่างละเอียดและเป็นขั้นตอนที่มีประสิทธิภาพ การเพิ่มทีแบบละเอียดนี้ทำให้สามารถสังเกตผลกระทบต่อค่า Training Loss และ Validation Loss ได้อย่างชัดเจน โดยไม่ทำให้การเปลี่ยนแปลงที่มากเกินไป สำหรับโมเดล MLP ที่มีข้อมูลแบบคงที่ซึ่งไม่มีความซับซ้อนและมีจำนวนไม่มาก การเพิ่มจำนวนหน่วยที่มากเกินไปอาจไม่จำเป็น เนื่องจากข้อมูลที่มีอยู่ไม่ต้องการความซับซ้อนในการประมวลผล การปรับขนาดหน่วยแบบละเอียดนี้ยังช่วยให้สามารถค้นหาจุดที่มีประสิทธิภาพสูงสุดในการประมวลผล โดยไม่ทำให้โมเดลซับซ้อนหรือสิ้นเปลืองทรัพยากรการประมวลผล

โดยสรุป เมื่อพิจารณาถึงผลลัพธ์ที่ได้จากการทดสอบ การเลือกใช้ 128 หน่วยถือว่าเป็นตัวเลือกที่เหมาะสมที่สุดสำหรับโมเดลนี้ เนื่องจากสามารถรักษาความเสถียรได้ดี ลดความผันผวนของ Validation Loss และช่วยให้โมเดลทั่วไปได้อย่างมีประสิทธิภาพ โดยไม่จำเป็นต้องเพิ่มจำนวนหน่วยเกินกว่านี้ ซึ่งจะช่วยประหยัดทรัพยากรในการประมวลผลและลดเวลาในการฝึกสอน ในขณะที่ยังคงรักษาประสิทธิภาพของโมเดลได้อย่างเต็มที่

3) L2 Regularizers

L2 Regularizers ด้วยค่าคงที่ 0.02 [129] ในโมเดล MLP มีบทบาทสำคัญในการลดการเกิด Overfitting ซึ่งเป็นสถานการณ์ที่โมเดลทำงานได้ดีมากกับข้อมูลฝึกอบรม การใช้ L2 Regularization ทำได้โดยการเพิ่มค่า Penalty ต่อพารามิเตอร์ของโมเดลที่มีค่าสูงเกินไป ซึ่งช่วยลดความซับซ้อนของโมเดลและบังคับให้โมเดลไม่พึ่งพาคุณสมบัติใดคุณสมบัติหนึ่งมากเกินไป การใช้ L2 Regularization Coefficient เป็นการกำหนดความแรงของ Penalty ที่จะบวกเข้าไปในฟังก์ชันการสูญเสีย หากค่า Coefficient สูงเกินไป โมเดลอาจบังคับมากเกินไปจนทำให้การเรียนรู้ของโมเดลไม่เพียงพอ แต่ถ้าค่า Coefficient ต่ำเกินไป การ Regularization อาจไม่มีผลเพียงพอในการป้องกัน Overfitting ค่า 0.02 ที่เลือกเพราะเป็นจุดสมดุลที่ดีระหว่างการรักษาความสามารถของโมเดลในการเรียนรู้จากข้อมูลฝึกอบรมและการลดความเสี่ยงของการ Overfitting ทำให้โมเดลสามารถทำนายข้อมูลใหม่ได้แม่นยำขึ้น.งานวิจัยนี้จึงกำหนดให้ L2 Regularizers เป็น 0.02

4) Optimizer

การเลือกใช้อัลกอริธึม Adam (Adaptive Moment Estimation) เป็นตัวปรับพารามิเตอร์ Optimizer ในโมเดล MLP มีความสำคัญมาก เนื่องจาก Adam [130] เป็นอัลกอริธึมที่รวมข้อดีของอัลกอริธึม Gradient Descent ที่ใช้อัตราการเรียนรู้ Learning Rate คงที่ช่วยเร่งความเร็วในการปรับพารามิเตอร์ในทิศทางที่ถูกต้อง การเลือก Adam มีข้อดีที่สำคัญอีกหลายประการ เช่น ความรวดเร็วและมีประสิทธิภาพสูงในด้านการคำนวณ ทำให้เหมาะสมกับการฝึกโมเดลที่มีขนาดใหญ่และข้อมูลจำนวนมาก นอกจากนี้ Adam ยังสามารถจัดการกับการเรียนรู้ที่มีการเปลี่ยนแปลงของ Gradient ได้

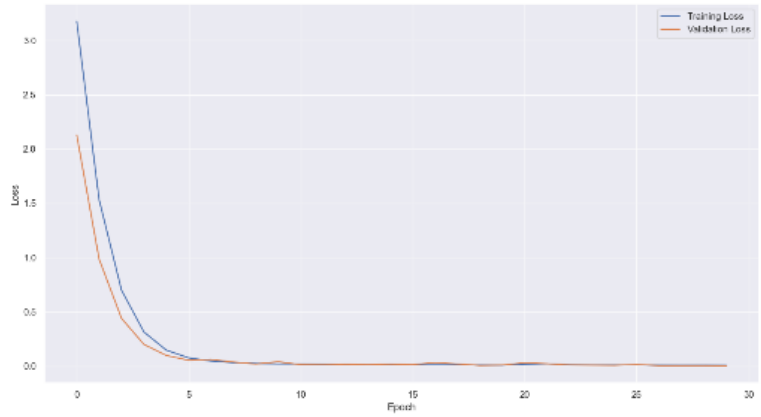
ดี ซึ่งทำให้โมเดลสามารถหลีกเลี่ยงปัญหาการติดอยู่ใน Local Minima ได้ดีกว่าอัลกอริธึมอื่น ๆ การใช้งาน Adam จึงเป็นที่นิยมอย่างมากในงานที่ต้องการความแม่นยำและความเสถียรสูงในการฝึกโมเดล MLP งานวิจัยนี้จึงกำหนด Optimizer เป็น Adam

5) Loss Function

Loss Function แบบ Cross Entropy ของโมเดล MLP ใช้ในการวัดความแม่นยำของการคาดการณ์ของโมเดลโดยเปรียบเทียบกับค่าจริงในการทำนาย โดยโมเดล MLP จะให้ค่าความน่าจะเป็นสำหรับแต่ละคลาส การคำนวณ Cross Entropy [131] จะพิจารณาว่าค่าความน่าจะเป็นที่โมเดลคาดการณ์นั้นใกล้เคียงกับค่าที่แท้จริงมากน้อยเพียงใด ซึ่งค่าที่แท้จริงจะเป็น 1 สำหรับคลาสที่ถูกต้องและ 0 สำหรับคลาสอื่นๆ หากโมเดลคาดการณ์ใกล้เคียงกับค่าที่แท้จริงมาก (เช่น ให้ความน่าจะเป็นสูงสำหรับคลาสที่ถูกต้อง) ค่า Cross Entropy จะต่ำ ซึ่งหมายความว่าโมเดลมีประสิทธิภาพดี แต่ถ้าโมเดลคาดการณ์ผิด การใช้ Cross Entropy ช่วยให้โมเดล MLP ปรับตัวได้ดีขึ้นโดยการพยายามลดค่าของ Loss Function นี้ให้ต่ำที่สุดระหว่างกระบวนการฝึก ซึ่ง Cross Entropy ได้ใช้อย่างแพร่หลายโดยสามารถอ้างอิงได้จาก [131]

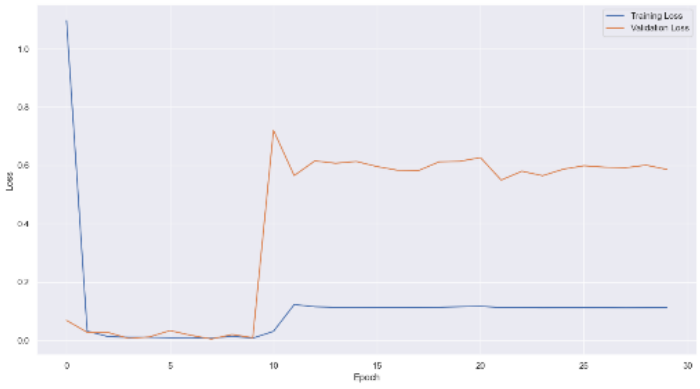
การกำหนดโครงสร้างภายนอกของโมเดล MLP นั้นจะประกอบไปด้วยสามขั้นตอนได้แก่ การกำหนดจำนวนรอบในการฝึก การกำหนด Batch Size และการกำหนด MLP เลเยอร์ โดยมีรายละเอียดดังต่อไปนี้

การกำหนดจำนวนรอบในการฝึกฝน (Epoch) โมเดล MLP นั้นสามารถพิจารณาได้จากภาพ 52 การลดลงของกราฟ Learning Curve สามารถสังเกตได้ว่าโมเดลเริ่มเข้าสู่สถานะ Convergence หรือความเสถียรประมาณที่ Epoch ที่ 10 หลังจากช่วงนี้ ค่า Loss ของทั้งสองส่วนลดลงอย่างค่อยเป็นค่อยไปและคงที่ในระดับต่ำ ซึ่งแสดงให้เห็นว่าโมเดลได้เรียนรู้และเข้าใจรูปแบบของข้อมูลที่ฝึกอย่างเพียงพอและไม่จำเป็นต้องปรับปรุงเพิ่มเติม ในจำนวนรอบในการฝึกฝนที่สูงขึ้นแม้ว่าค่า Loss จะยังคงลดลงเล็กน้อยหลังจากจำนวนรอบในการฝึกฝนที่ 10 แต่การเปลี่ยนแปลงนั้นเกิดขึ้นในระดับที่น้อยมาก โดยค่า Loss ของ Validation และ Training เริ่มคงที่ แสดงให้เห็นถึงการเข้าสู่สถานะ Convergence ที่ดีของโมเดล ในกรณีนี้ การรันถึง 30 Epoch ถือว่าเป็นการตัดสินใจที่เหมาะสม เพราะช่วยให้มั่นใจได้ว่าโมเดลมีความเสถียรและสามารถทำงานได้อย่างมีประสิทธิภาพเมื่อนำไปใช้งานจริง โดยค่า Loss ที่อยู่ในระดับต่ำสุดและคงที่บ่งบอกถึงความสามารถในการทำนายที่ดีโดยไม่เกิดการ Overfitting ของข้อมูล

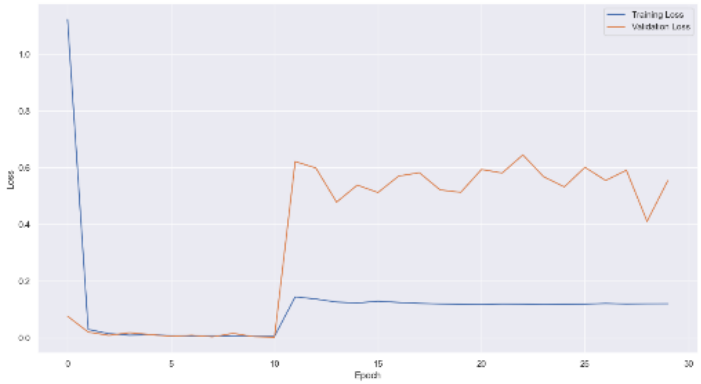


ภาพ 52 จำนวนรอบที่เหมาะสมในการฝึกฝนของโมเดล MLP

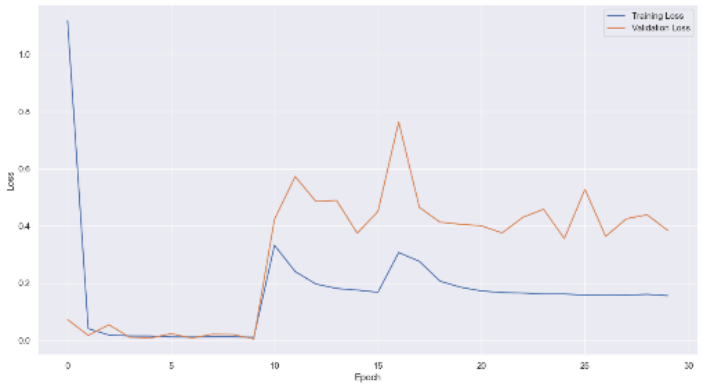
การกำหนด Batch Size ของโมเดล MLP นั้นจะทำการกำหนด Batch Size โดยการสุ่มปรับขนาดของ Batch Size ที่เหมาะสมที่สุดของโมเดล MLP โดยการหาค่าที่ดีที่สุดจะอธิบายในขั้นตอนการปรับ Batch Size โดยการกำหนดแบบไล่ขนาดตั้งแต่ 10 จนถึง 60 โดยผลลัพธ์ของการทดลองการปรับเพื่อหา Batch Size ที่ดีที่สุดของโมเดล MLP แสดงในภาพ 53-58



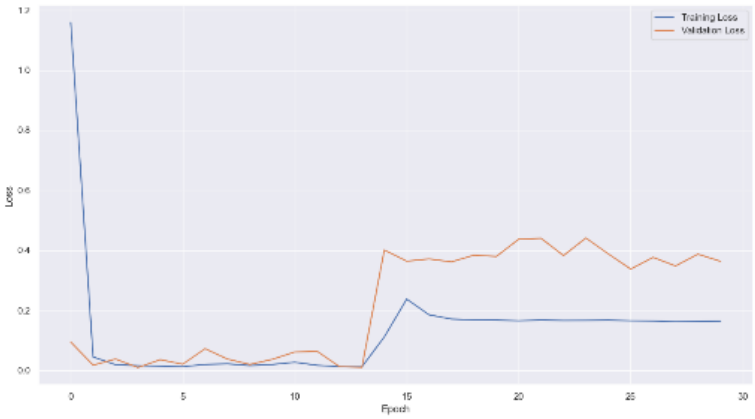
ภาพ 53 กำหนด Batch Size ที่ 10 ของโมเดล MLP



ภาพ 54 กำหนด Batch Size ที่ 20 ของโมเดล MLP



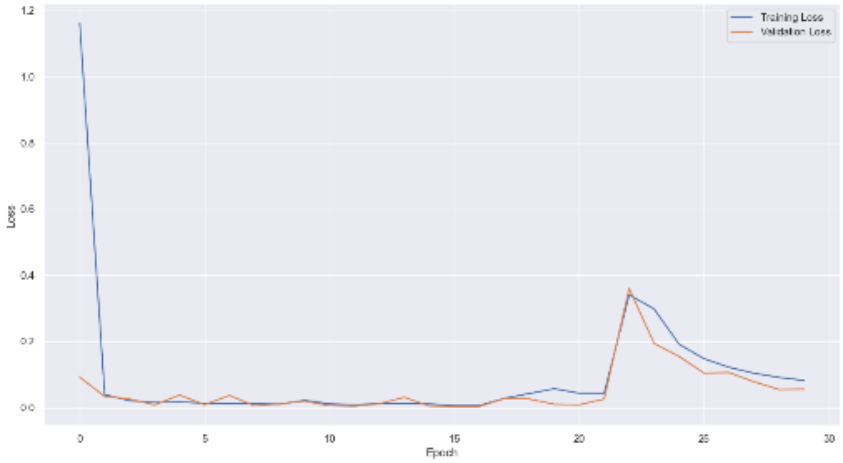
ภาพ 55 กำหนด Batch Size ที่ 30 ของโมเดล MLP



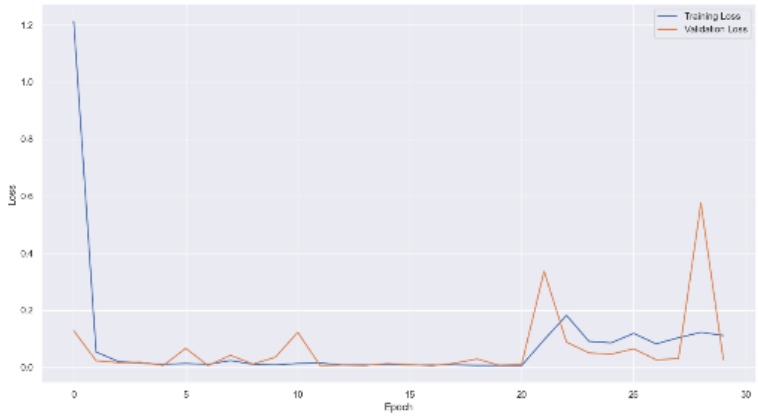
ภาพ 56 กำหนด Batch Size ที่ 40 ของโมเดล MLP

จากการเปรียบเทียบผลการเรียนรู้ของ MLP โดยใช้ Batch Size ต่าง ๆ ตั้งแต่ 10 20 30 และ 40 พบว่าการลดลงของค่า Training Loss ในทุกกรณีเป็นไปอย่างรวดเร็วและคงที่หลังจากช่วงแรกของการฝึกสอน อย่างไรก็ตาม ค่า Validation Loss ในแต่ละกรณีมีความผันผวน โดยเฉพาะในช่วงหลังของการฝึกสอน ซึ่งบ่งชี้ถึงความเสี่ยงในการเกิด Overfitting ในโมเดล

เมื่อใช้ Batch Size ที่เล็กกว่า เช่น ขนาด 10 ดังแสดงในภาพ 53 และขนาด 20 ดังแสดงในภาพ 54 ค่า Validation Loss มีความผันผวนที่ค่อนข้างสูง แสดงให้เห็นว่าโมเดลอาจไม่สามารถทั่วไปกับข้อมูลที่ไม่เคยเห็นมาก่อน แม้ว่า Batch Size 20 จะลดความผันผวนลงเล็กน้อย แต่ยังคงมีแนวโน้มที่จะเกิด Overfitting อย่างชัดเจน ซึ่งแสดงถึงความไม่เสถียรในการทำนายข้อมูลที่ไม่อยู่ในชุดฝึกสอน เมื่อเพิ่ม Batch Size ไปที่ขนาด 30 ดังแสดงในภาพ 55 และขนาด 40 ดังแสดงในภาพ 56 ความผันผวนของค่า Validation Loss ลดลง แสดงถึงแนวโน้มที่โมเดลมีความเสถียรมากขึ้น ขนาด Batch Size ที่ใหญ่ขึ้นช่วยให้การเรียนรู้มีความเสถียรและคงที่มากขึ้น ลดการเปลี่ยนแปลงของ Validation Loss ในแต่ละ Epoch แม้ว่าความเสี่ยงในการเกิด Overfitting ยังคงมีอยู่ การใช้ Batch Size ขนาดใหญ่จึงเป็นปัจจัยหนึ่งที่สามารถช่วยปรับปรุงเสถียรภาพของโมเดลได้



ภาพ 57 กำหนด Batch Size ที่ 50 ของโมเดล MLP

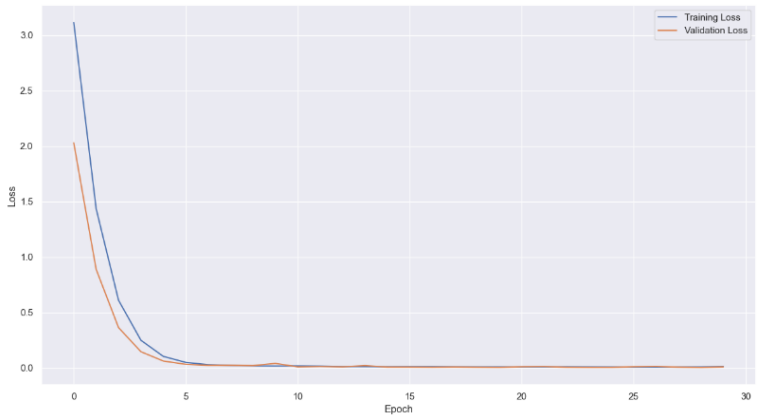


ภาพ 58 กำหนด Batch Size ที่ 60 ของโมเดล MLP

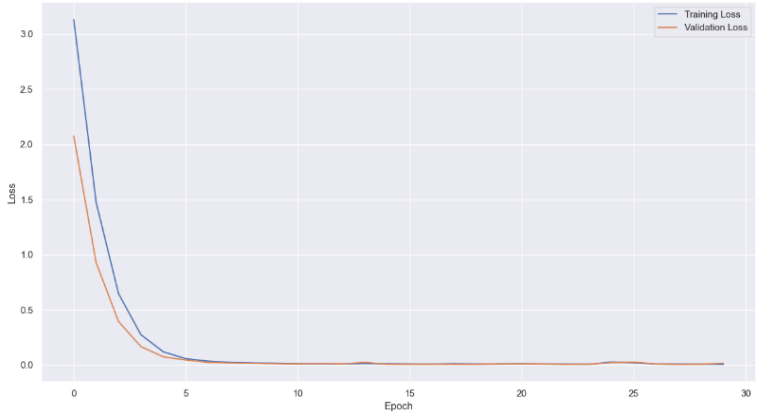
จากการเปรียบเทียบผลการเรียนรู้ของ MLP โดยใช้ Batch Size ที่ 40 50 และ 60 สามารถเห็นความแตกต่างในเสถียรภาพของโมเดลได้อย่างชัดเจน โดยแต่ละขนาดของ Batch Size มีผลต่อการเรียนรู้และค่าความสูญเสีย (loss) ในลักษณะต่าง ๆ

เริ่มต้นที่ Batch Size 40 พบว่าค่า Training Loss ลดลงอย่างรวดเร็วและคงที่ตลอดการฝึกสอน แต่ค่า Validation Loss ยังคงมีความผันผวนอย่างชัดเจน โดยเฉพาะในช่วงท้ายของการฝึกสอน ความผันผวนนี้บ่งชี้ถึงการที่โมเดลอาจยังไม่เสถียรเพียงพอ และมีความเสี่ยงต่อการ Overfitting อยู่ในระดับหนึ่ง ต่อมาเมื่อใช้ Batch Size 50 ผลลัพธ์แสดงให้เห็นถึงความเสถียรที่เพิ่มขึ้นดังแสดงในภาพ 57 ค่า Training Loss และ Validation Loss มีการลดลงอย่างสม่ำเสมอและคงที่มากขึ้นเมื่อเทียบกับ Batch Size 40 ความผันผวนของค่า Validation Loss ลดลงอย่างมาก ทำให้โมเดลมีความเสถียรและสามารถทั่วไปได้ดีขึ้นกับข้อมูลที่ไม่เคยเห็นมาก่อน สุดท้ายที่ Batch Size 60 แม้ว่าค่า Training Loss จะลดลงได้อย่างรวดเร็วและคงที่ดังแสดงในภาพ 58 เช่นเดียวกับ Batch Size 50 แต่ค่า Validation Loss กลับมีความผันผวนที่สูงขึ้นเมื่อเทียบกับ Batch Size 50 โดยเฉพาะในช่วงท้ายของการฝึกสอน ซึ่งแสดงว่า Batch Size ที่ใหญ่กว่า 50 อาจไม่ให้ประโยชน์เพิ่มเติมในด้านเสถียรภาพ แต่กลับทำให้โมเดลมีแนวโน้มที่จะเกิดความผันผวนมากขึ้น สรุปได้ว่า Batch Size 50 เป็นขนาดที่เหมาะสมที่สุดในกรณีนี้ เนื่องจากให้เสถียรภาพดีและลดความผันผวนของค่า Validation Loss ได้อย่างมีประสิทธิภาพเมื่อเทียบกับขนาดอื่น ๆ อย่างไรก็ตามการปรับ Batch Size ในขนาดต่างๆนั้นก็ยังพบว่ามีการเกิด Overfitting อยู่ ดังนั้นจึงจำเป็นที่จะต้องปรับพารามิเตอร์ตัวอื่นเพื่อให้โมเดลนั้นเกิดความเสถียร ผู้วิจัยจึงกำหนด Batch Size ของโมเดล MLP คือ 50

การกำหนด MLP เลเยอร์ เป็นการทดลองการเพิ่มเลเยอร์ของโมเดล MLP เพื่อให้เหมาะสมและสอดคล้องกับข้อมูลแบบคงที่ที่มีอยู่ โดยโมเดล MLP นั้นจะใช้จำนวนเลเยอร์เพียง 1 เลเยอร์ก็สามารถทำงานได้ดีเนื่องจากข้อมูลในส่วนของโมเดล MLP นั้นไม่ซับซ้อนมากเมื่อเทียบกับโมเดล LSTM ที่ข้อมูลนั้นมีความซับซ้อนจึงต้องสร้างเลเยอร์ให้เหมาะสมกับการเรียนรู้ของข้อมูล ผลการเพิ่มเลเยอร์ของโมเดล MLP แสดงในภาพ 59-60



ภาพ 59 กำหนดโมเดล MLP แบบ 1 เลเยอร์



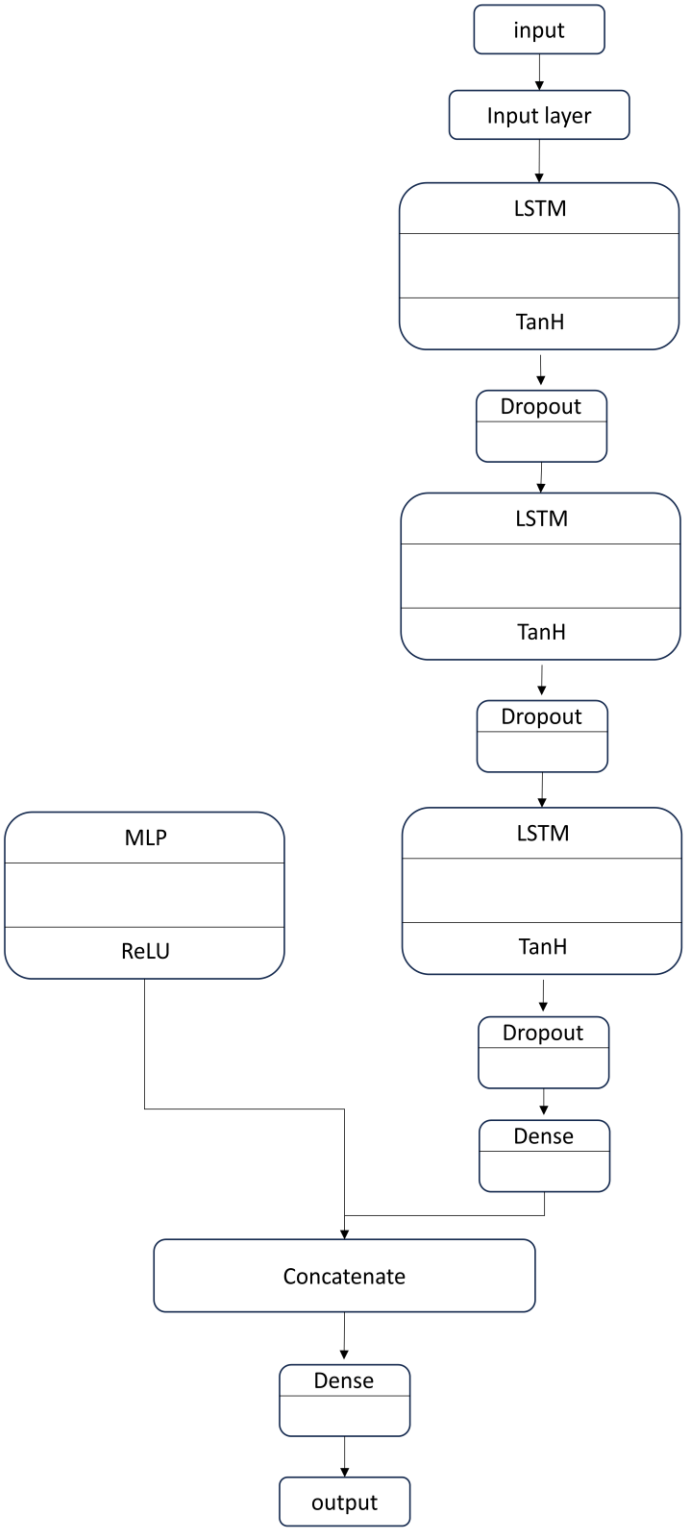
ภาพ 60 กำหนดโมเดล MLP แบบ 2 เลเยอร์

จากการเปรียบเทียบผลการเรียนรู้ของ MLP โดยใช้จำนวนเลเยอร์ที่แตกต่างกันในภาพ 56 ซึ่งใช้ 1 เลเยอร์ และภาพ 57 ซึ่งใช้ 2 เลเยอร์ พบว่าค่า Training Loss และ Validation Loss ในทั้ง

สองกรณีลดลงอย่างสม่ำเสมอและคงที่ ทำให้เห็นได้ว่าโมเดลมีความเสถียรในระดับใกล้เคียงกันโดยไม่คำนึงถึงจำนวนเลเยอร์ที่ใช้

อย่างไรก็ตาม การเพิ่มเลเยอร์ในโมเดลจาก 1 เลเยอร์เป็น 2 เลเยอร์ไม่ได้แสดงถึงประโยชน์ที่เพิ่มขึ้นอย่างมีนัยสำคัญในแง่ของความเสถียรหรือความสามารถในการทำนาย ขณะที่ทั้งสองกรณีแสดงให้เห็นว่าค่า Loss ลดลงได้อย่างรวดเร็วและคงที่ การเพิ่มเลเยอร์เพิ่มเติมอาจไม่จำเป็นสำหรับการใช้งานทั่วไป เนื่องจาก 1 เลเยอร์เพียงพอต่อการประมวลผลและให้ผลลัพธ์ที่น่าพอใจ นอกจากนี้ การใช้เพียง 1 เลเยอร์ยังช่วยลดการใช้ทรัพยากรในการประมวลผล ทำให้โมเดลมีความประหยัดและสามารถนำไปใช้งานในสถานการณ์ที่ต้องการการคำนวณอย่างมีประสิทธิภาพโดยไม่ต้องเสียทรัพยากรเพิ่มเติมในการฝึกสอนหรือทำนายข้อมูล

การออกแบบสถาปัตยกรรมของโมเดลที่ประกอบด้วย LSTM และ MLP ดังแสดงในภาพ 61 ที่ได้จากการปรับปรุงในขั้นตอนก่อนหน้า ซึ่งทำงานร่วมกันเพื่อเพิ่มประสิทธิภาพในการประมวลผลข้อมูล กระบวนการเริ่มต้นจากข้อมูลนำเข้าผ่านชั้น Input เลเยอร์โดยข้อมูลที่ได้รับจะส่งผ่านไปยังชั้น LSTM 3 ชั้น ซึ่งในแต่ละชั้นของ LSTM จะใช้ Tanh Activation Function ในการประมวลผลข้อมูล LSTM เป็นโครงสร้างที่เหมาะสมสำหรับการจัดการกับข้อมูลอนุกรมเวลาโดยเฉพาะ และการใช้ 3 ชั้นของ LSTM ช่วยให้โมเดลสามารถจับข้อมูลที่ซับซ้อนได้อย่างมีประสิทธิภาพ หลังจากผ่านการประมวลผลในแต่ละชั้นของ LSTM ข้อมูลจะส่งผ่านชั้น Dropout ซึ่งมีบทบาทในการลดความซับซ้อนของโมเดลเพื่อลดปัญหา Overfitting โดยการสุ่มปิดบางหน่วยประมวลผลระหว่างการฝึกอบรม เมื่อข้อมูลผ่านการประมวลผลจากชั้น LSTM แล้ว ข้อมูลจะส่งผ่านไปยังชั้น Dense ซึ่งจะทำหน้าที่ในการรวมข้อมูล (Concatenate) ที่ได้รับจาก MLP ที่ได้ผ่านกระบวนการการประมวลผลของตัวเองด้วย ReLu Activation Function ข้อดีของ ReLu Activation Function คือ ประมวลได้รวดเร็วเนื่องจากสนใจเฉพาะข้อมูลที่มีค่ามากกว่า 0 [132] การรวมข้อมูลจากทั้ง LSTM และ MLP เข้าด้วยกันช่วยให้โมเดลสามารถประมวลผลข้อมูลได้ทั้งในเชิงเวลาและคงที่ ทำให้การทำนายมีความแม่นยำมากขึ้น สุดท้ายข้อมูลที่รวมแล้วจะส่งผ่านชั้น Dense ก่อนที่จะส่งออกผลลัพธ์ผ่าน Output เลเยอร์ การกำหนดพารามิเตอร์ของโมเดล LSTM + MLP แสดงในตาราง 6



ภาพ 61 โครงสร้างโมเดล LSTM + Multilayer Perceptron

ตาราง 6 การกำหนดค่าในการสร้างโมเดล

หน่วยย่อยของโครงสร้างของ LSTM	ค่าที่กำหนด	หน่วยย่อยของโครงสร้างของ MLP	ค่าที่กำหนด
หน่วยย่อยใน LSTM เลเยอร์		**หน่วยย่อยใน MLP เลเยอร์**	
feature_range	(-1,1)	Validation split	0.1
values.reshape	(-1,1)	Activation function	ReLU
model	Sequential	Unit	128
LSTM Unit	600	Learning rate	0.001
activation function	TanH	Regularizers.l2	0.02
Regularizers l2	0.01	optimizers	Adam
optimizer	Adam	loss function	Cross entropy
loss function	MSE	**หน่วยย่อยนอก MLP เลเยอร์**	
Validation split	0.1	epoch	20
หน่วยย่อยนอก LSTM เลเยอร์		Batch size	50
drop out	0.25		
epoch	60		
Batch Size	720		

ตารางที่ 6 สรุปละเอียดเกี่ยวกับพารามิเตอร์ที่กำหนดในการพัฒนาและปรับปรุงโมเดลสำหรับการทำนาย โดยใช้โมเดล LSTM + MLP ตารางนี้สรุปพารามิเตอร์สำคัญที่ได้รับการปรับแต่งเพื่อหาประสิทธิภาพที่ดีที่สุดของโมเดล LSTM + MLP โดยพารามิเตอร์เหล่านี้มีบทบาทสำคัญในการกำหนดโครงสร้างและการทำงานของโมเดลให้เหมาะสมกับข้อมูล

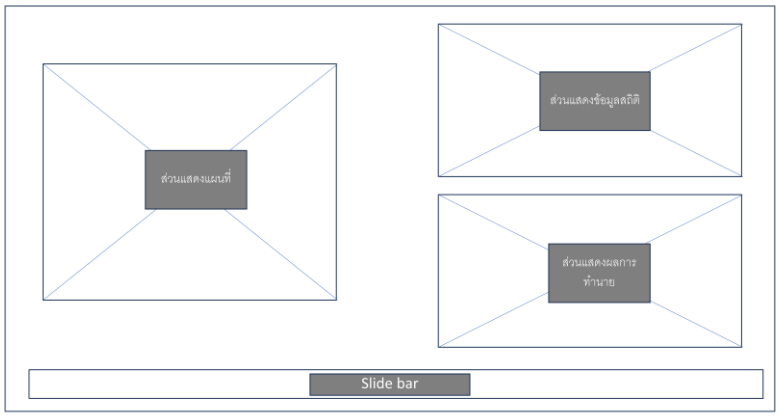
เมื่อได้โมเดล LSTM ที่สามารถทำงานร่วมกันกับโมเดล MLP ได้อย่างมีประสิทธิภาพแล้วนั้น โมเดลนี้จะนำไปทดสอบผ่านการวัดประสิทธิภาพของโมเดลสำหรับการทำนายโดยผ่านการวัดความแม่นยำผ่านเมตริกส์การวัดด้วย RMSE และ MAE ของชุดข้อมูลจริงและชุดข้อมูลการทดสอบเพื่อแสดงให้เห็นถึงประสิทธิภาพของการทำนาย โดยผลการทำนายของโมเดล LSTM ที่สามารถทำงานร่วมกันกับโมเดล MLP

3.5 โมดูลที่ 3 การสร้างเว็บแอปพลิเคชัน

เมื่อผลลัพธ์จากการสร้างโมเดลพล็อตในแอปพลิเคชันเว็บที่สร้างขึ้นเพื่อแสดงผลลัพธ์ของโมเดล จะแสดงตำแหน่งภูมิภาคของแต่ละจังหวัดในประเทศไทย และในแต่ละจังหวัดจะมีการแสดงผลลัพธ์การทำนายของโมเดลในรูปแบบกราฟและแสดงค่า MAE RMSE ที่ได้จากผลลัพธ์ของ

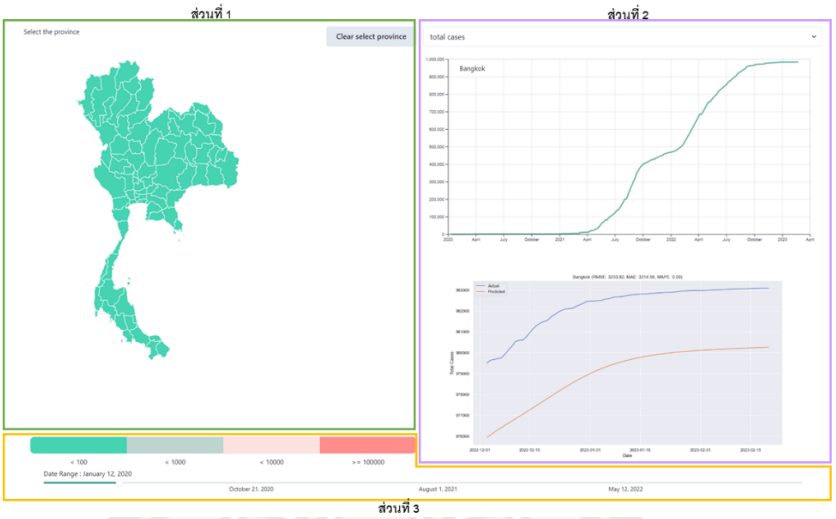
การทำนายจากโมเดลที่ได้พัฒนาขึ้น และมีแถบเลื่อนหรือ Slide bar เพื่อช่วยให้ติดตามข้อมูลในแต่ละช่วงได้สะดวกมากขึ้น

ในการสร้างแผนที่ของประเทศไทย ใช้ไลบรารี React for Web และนำแผนที่มาผสมกับผลลัพธ์การทำนาย โควิด-19 ให้ได้ระบบข้อมูลของแต่ละจังหวัดในประเทศไทย เช่น เมื่อผู้ใช้เลือกจังหวัดใดๆ บนแผนที่ จะแสดงป๊อปอัพ (Pop-Up) ที่แสดงผลการทำนายและความแม่นยำที่เกี่ยวข้อง MAE RMSE สำหรับกราฟและประกอบด้วยแผนที่ที่แสดงผลการทำนายแบบละเอียด



ภาพ 62 การออกแบบโครงสร้างเว็บแอปพลิเคชั่น

ภาพ 62 แสดงการออกแบบ Layout ของแอปพลิเคชันเว็บสำหรับแสดงผลการทำนายโควิด-19 ซึ่งประกอบด้วยส่วนแสดงแผนที่ประเทศไทยด้านซ้ายที่ผู้ใช้สามารถเลือกจังหวัดเพื่อดูผลการทำนาย ข้อมูลสถิติผู้ติดเชื้อในรูปแบบกราฟด้านขวาบน และผลการทำนายจากโมเดลที่รวมถึงค่า MAE และ RMSE ด้านขวาล่าง โดยแผนที่สร้างด้วยไลบรารี React for web ทำให้ผู้ใช้เห็นข้อมูลการทำนายและสถิติของแต่ละจังหวัดเมื่อเลือกบนแผนที่ ส่วนแถบเลื่อนด้านล่างช่วยในการเลื่อนดูข้อมูลเพิ่มเติม ภาพ 63 เป็นผลลัพธ์ที่แสดงข้อมูลตามการออกแบบนี้

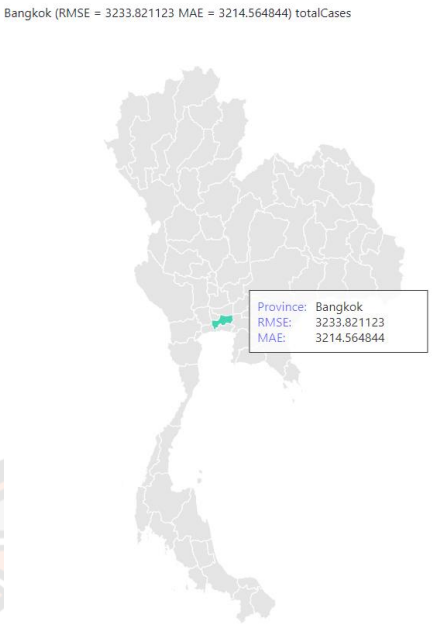


ภาพ 63 เว็บแอปพลิเคชั่นที่พัฒนาขึ้นโดยแบ่งออกเป็น 3 ส่วน

กระบวนการสร้างแผนที่ประกอบด้วยสามส่วนดังนี้ ส่วนที่ 1 แผนที่ ส่วนที่ 2 กราฟสถิติของข้อมูล และ ส่วนที่ 3 ผลการทำนาย โดยมีรายละเอียดดังต่อไปนี้

1) ส่วนที่ 1 แผนที่

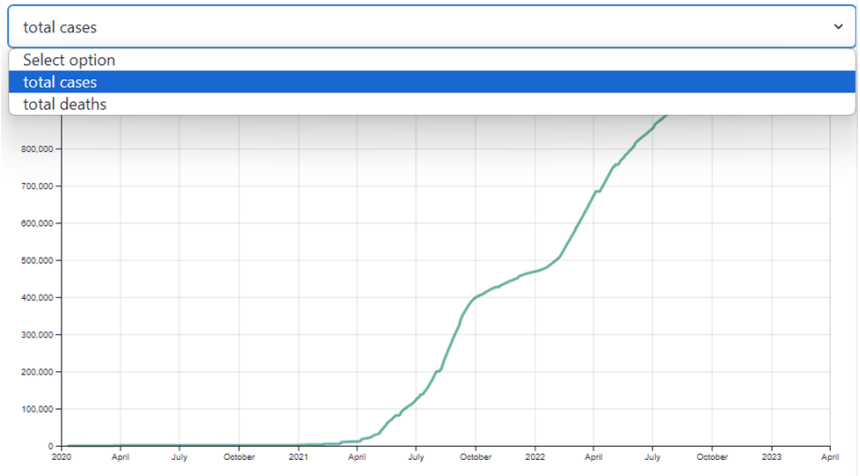
การสร้างแผนที่ประเทศไทยในส่วนที่ 1 ดังภาพ 64 นั้นสร้างโดยการใช้ไลบรารี่ของ React js โดยการใช้พิกัดของแต่ละจังหวัดในประเทศไทยที่จัดเก็บอยู่ในรูปแบบของ Json โดยภายในนั้นจะประกอบไปด้วย ละติจูดและลองติดจูดของแต่ละจังหวัด นอกจากจะแสดงผลแผนที่ของประเทศไทยแล้วนั้นส่วนนี้ยังจะแสดงผลการทำนายที่อยู่ในรูปแบบของ RMSE และ MAE ที่ได้มาจากการทำนายของโมเดลที่จัดเก็บภายในไฟล์ .csv ผู้ใช้จำเป็นต้องเลือกจังหวัดที่ต้องการทราบผลการทำนายก่อนจึงจะแสดงผลการทำนายของจังหวัดนั้นๆปรากฏเป็น Pop Up อยู่บนเขตพื้นที่ของจังหวัดที่เลือก



ภาพ 64 ผลลัพธ์ของการสร้างแผนที่ประเทศไทย

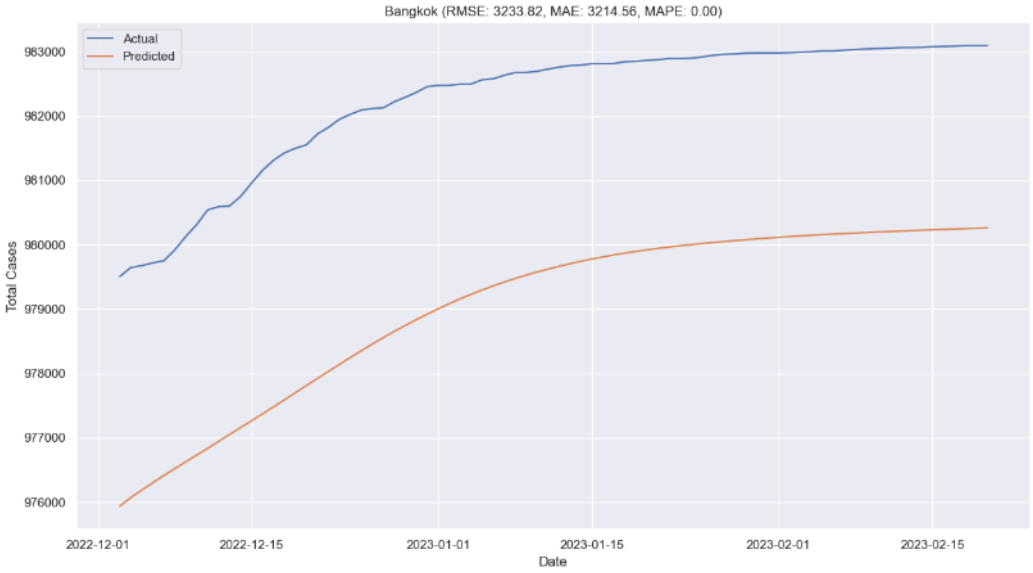
2) ส่วนที่ 2 กราฟสถิติของข้อมูล

การแสดงผลในรูปแบบกราฟของส่วนที่ 2 นี้เป็นการแสดงผลในรูปแบบของกราฟของข้อมูลจริงร่วมกับผลการทำนายที่ได้จากการพัฒนาโมเดล โดยส่วนที่ 2 กราฟสถิติของข้อมูล ประกอบไปด้วย 2 กราฟและการแสดงผลของทั้งสองกราฟนั้นมีความสัมพันธ์กันในเรื่องของข้อมูล โดยในแต่ละกราฟแสดงข้อมูลดังภาพ 65



ภาพ 65 การเลือกข้อมูลที่จะแสดงผลบนแผนที่

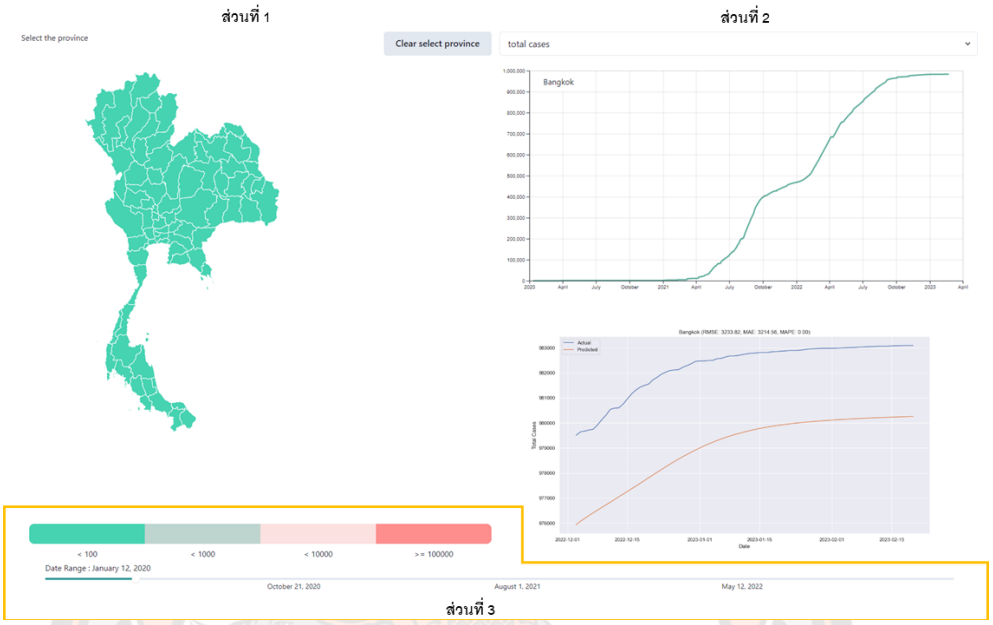
กราฟที่ 1 ดังงภาพ 65 จะแสดงข้อมูลจริงที่พล็อตจากข้อมูลที่ได้ทำผ่านกระบวนการการประมวลผลข้อมูลล่วงหน้าที่อยู่ในไฟล์ .csv ของแต่ละจังหวัดโดยมีระยะการแสดงข้อมูลเริ่มต้นที่ วันที่ 12 มกราคม 2563 จนถึง 2/20/2566 และแถบด้านบนสุดผู้ใช้สามารถเลือกการแสดงผลที่แยกกันระหว่างการทำนายจำนวนผู้ติดเชื้อสะสมและการทำนายจำนวนผู้เสียชีวิตสะสมได้



ภาพ 66 การทำนายของโมเดลบนเว็บแอปพลิเคชัน

กราฟที่ 2 ดังแสดงภาพ 66 เป็นการแสดงผลของข้อมูลจริงร่วมกับผลลัพธ์ที่ได้จากการทำนายที่ได้จากโมเดลโดยภายในกราฟจะประกอบไปด้วยชุดข้อมูล 2 เชิงเส้น ประกอบไปด้วย เส้นข้อมูลจริงและเส้นข้อมูลที่เกิดจากการทำนายของโมเดลและภายในด้านบนของกราฟที่ 2 ก็แสดงถึงผลของ RMSE และ MAE ด้วยเช่นกัน

3) ส่วนที่ 3 ผลการทำนาย



ภาพ 67 แถบเลื่อนเพื่อดูสถิติของการแพร่ระบาดของเชื้อโควิด-19

ส่วนที่ 3 เป็นส่วนที่ใช้สำหรับการแสดงผลการติดตามผลการติดเชื้อสะสมนับตั้งแต่วันแรกที่มีการยืนยันการติดเชื้อผู้ติดเชื้อรายแรกในประเทศไทยดังแสดงในภาพ 67 ส่วนนี้จะสร้างโดยการใช้ข้อมูลจริงของทุกจังหวัดมาแสดงผลบนแผนที่ตามพิกัดของแต่ละจังหวัดส่วนที่ 3 นี้จัดทำขึ้นเพื่อเพิ่มความหลากหลายในการแสดงผลการติดตามสถานการณ์การติดเชื้อโควิด-19 ให้แก่ผู้ใช้

บทที่ 4
ผลการทดลอง

การรายงานผลการทดลองของการพัฒนาโมเดลเพื่อทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 นั้นแบ่งออกเป็น 2 ชุดการทดสอบ ได้แก่ ผลการทดสอบการทำนายโดยการเปรียบเทียบระหว่างข้อมูลจริงในรูปแบบอนุกรมเวลา โดยข้อมูลจริงนั้นจะใช้เพื่อวัดความแม่นยำและประสิทธิภาพของโมเดลที่ได้พัฒนาขึ้นสำหรับการทำนาย โดยใช้เมตริกการวัดคือ RMSE (Root Mean Squared Error) และ MAE (Mean Absolute Error) และผลการทดสอบประสิทธิภาพของโมเดลโดยการใช้ Learning Curve เพื่อวิเคราะห์การเรียนรู้และประสิทธิภาพของโมเดล ซึ่งผลลัพธ์ของการทดสอบเหล่านี้จะนำไปสู่การสร้างแอปพลิเคชันเว็บที่ใช้สำหรับแสดงผลการทำนาย โดยผลลัพธ์การสร้างเว็บแอปพลิเคชันนั้นจะรายงานไว้ในบทที่ 4 ซึ่งต่อจากผลการทดสอบของโมเดลที่ใช้สำหรับการทำนายเพื่อให้ผู้ใช้สามารถเห็นภาพรวมของการทำนายและประสิทธิภาพของโมเดลได้อย่างชัดเจน

4.1 ผลการพัฒนาโมเดล LSTM แบบพื้นฐาน

จากผลการทดลองที่ได้จากการพัฒนาโมเดลสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด- 19 โดยการใช้ โมเดล LSTM แบบพื้นฐานและ LSTM + MLP โดยใช้เมตริกในการทดสอบประสิทธิภาพของโมเดลได้แก่ RMSE MAE MSE นอกจากนั้นยังเพิ่มการวิเคราะห์ประสิทธิของโมเดลโดยการใช้การ Learning Curve เพื่อสังเกตุการแนวโน้มของประสิทธิของโมเดล โดยการอภิปรายผลการทดลองคือ การวิเคราะห์ประสิทธิภาพของโมเดลโดยการใช้ Learning Curve

4.1.1 การประเมินประสิทธิภาพของโมเดลในการฝึกอบรมโมเดล Model Validation

การวิเคราะห์ประสิทธิภาพของโมเดลในกระบวนการฝึกอบรมโมเดล จะการสร้างเส้นกราฟของค่าของ Learning Curve เพื่อให้เห็นการเปลี่ยนแปลงของค่า Loss ขณะการฝึกอบรมและการทดสอบบนชุดข้อมูลที่ไม่ได้ใช้ในการฝึก (Validation) ตามลำดับ ดังนี้ การวิเคราะห์ประสิทธิภาพของโมเดลในกระบวนการฝึกอบรมสามารถทำได้โดยการสร้างกราฟเส้นของค่า Learning Curve เพื่อให้เห็นการเปลี่ยนแปลงของค่าสูญเสียขณะการฝึกอบรมและการทดสอบบนชุดข้อมูลที่ไม่ได้ใช้ในการฝึก (Validation) ตามลำดับ ดังต่อไปนี้

- Training Loss Curve คือการแสดงค่าของฟังก์ชัน Loss ของโมเดลในแต่ละรอบการฝึก (Epoch) ของชุดข้อมูลการฝึก ซึ่งใช้เพื่อวัดความผิดพลาดของโมเดลในการทำนายผลลัพธ์ตามข้อมูลเป้าหมาย (Ground Truth) ในชุดข้อมูลการฝึกเรียนรู้และปรับค่าพารามิเตอร์

ต่างๆ เพื่อให้ผลลัพธ์มีความใกล้เคียงกับเป้าหมาย ค่า Training Loss มีแนวโน้มจะลดลงในแต่ละรอบการฝึก และอาจเข้าสู่ตัวเลขเล็กสุดหรือเป็นค่าคงที่ในที่สุด

- Validation Loss Curve (เส้นกราฟค่า Validation Loss) นั้นเป็นเส้นกราฟที่แสดงค่าของฟังก์ชัน Loss ของโมเดลในแต่ละรอบการฝึก โดยที่ข้อมูลที่ใช้ในการคำนวณค่า Loss เป็นชุดข้อมูลการทดสอบที่ไม่เคยใช้ในการฝึก (Validation Dataset) ซึ่งข้อมูลนี้มีไว้เพื่อตรวจสอบว่าโมเดลสามารถทำนายข้อมูลที่ไม่เคยเห็นก่อนได้อย่างไร ค่า Validation Loss จะเพิ่มขึ้นเมื่อโมเดลมีการเรียนรู้เกินไปและเกิดการต่อ Overfitting (การเรียนรู้เกินขนาด)

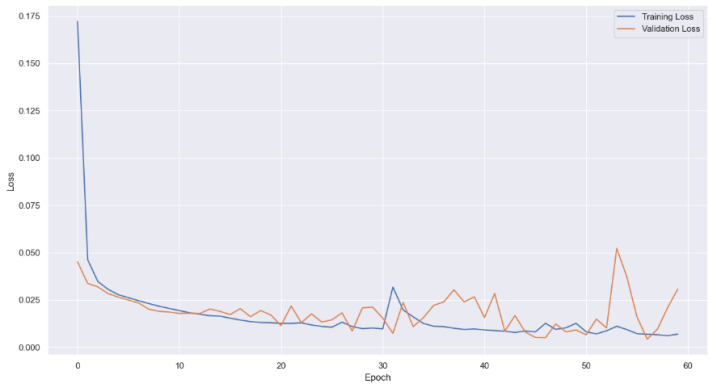
ตัวอย่างการวิเคราะห์ประสิทธิภาพโมเดล LSTM แบบพื้นฐานโดยใช้ Learning Curve ที่เป็นการพิจารณาระหว่างเส้นของ Training Loss และ Validation Loss ของจังหวัดกรุงเทพมหานครและเป็นตัวอย่างของ Training Loss และ Validation Loss ที่เห็นความแตกต่างและเห็นความไม่แม่นยำของโมเดลได้อย่างชัดเจน

ภาพ 68 ตัวอย่าง Learning Curve ของโมเดล LSTM แบบพื้นฐานโดยใช้ชุดข้อมูลการติดเชื้อสะสมของจังหวัดกรุงเทพมหานคร

จากภาพ 68 ในการประเมินประสิทธิภาพโมเดลนั้นพบว่ากราฟของ Training loss มีแนวโน้มที่ลดลงจากการฝึกอบรมของโมเดล LSTM แบบพื้นฐานแต่รูปแบบของกราฟ Training loss นั้นยังไม่เข้าสู่สถานะเสถียรหรือสถานะที่ดีที่สุด ในส่วนของกราฟ Validation loss นั้นพบว่ามีค่า loss ที่ไม่คงที่และมีแนวโน้มที่เพิ่มขึ้นอีกทั้งยังไม่มีการเข้าสู่สถานะเสถียรเช่นกัน เมื่อพิจารณา training loss และ validation loss เข้าด้วยกันพบว่ากราฟทั้งสองนั้นไม่อยู่ในระนาบเดียวกันและไม่มีแนวโน้มที่จะมาบรรจบกันทำให้ทั้งสองกราฟนั้นไม่เข้าสู่สถานะ Convergence ซึ่งเป็นผลให้โมเดล

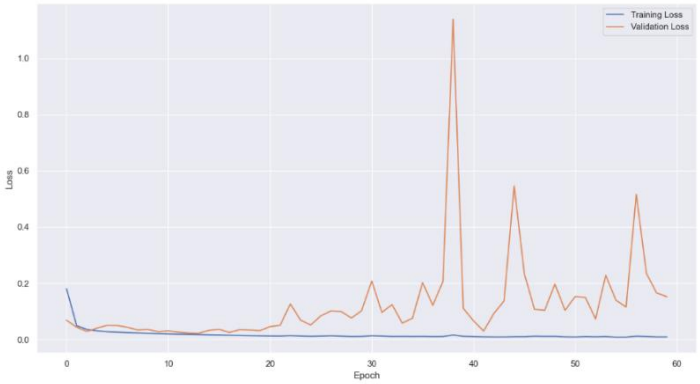
LSTM แบบพื้นฐานมีความไม่เสถียรและไม่สามารถใช้ในการทำนายสถานการณ์การแพร่เชื้อโควิด -19 ได้

เมื่อพิจารณากราฟของ Training Loss และ Validation Loss ของทุกจังหวัดแล้วนั้นพบว่ารูปแบบของกราฟ Training Loss และ Validation Loss ของทุกจังหวัดมีลักษณะของกราฟใกล้เคียงกันคือ กราฟของ Training Loss นั้นมีแนวโน้มที่ลดจากการฝึกฝนของโมเดลแต่จะมีความไม่คงที่ของเส้นกราฟและไม่เข้าสู่สถานะเสถียรของโมเดล ในส่วนของ Validation Loss นั้นพบว่ามีแนวโน้มลดลงแต่ไม่มีความคงที่และไม่เข้าสู่สถานะเสถียรเช่นกัน และเมื่อพิจารณาทั้งสองกราฟเข้าด้วยกันแล้วพบว่าทั้งสองกราฟนั้นมีแนวโน้มที่บรรจบกันแต่เป็นการบรรจบกันที่ไม่เข้าสู่สถานะเสถียรเนื่องจากรูปแบบของกราฟทั้งสองนั้นมีแนวโน้มไม่คงที่



ภาพ 69 ตัวอย่าง Learning Curve ของโมเดล LSTM แบบพื้นฐานโดยใช้ชุดข้อมูลการเสียชีวิตสะสมของจังหวัดฉะเชิงเทรา

จากภาพ 69 จากกราฟของ Training Loss นั้นมีแนวโน้มที่ลดลงจากการฝึกฝนของโมเดลแต่เมื่อเข้าใกล้รอบการฝึกฝนที่ 30 นั้นพบว่ากราฟของ Training Loss นั้นมีแนวโน้มที่เพิ่มขึ้นและลดลงในรอบของการฝึกฝนถัดไปจึงทำให้กราฟไม่คงที่และไม่เข้าสู่สถานะเสถียร และเมื่อพิจารณาในส่วนของกราฟ Validation Loss นั้นพบว่าเกิดความไม่เสถียรด้วยเช่นกันจึงเป็นผลทำให้เมื่อทำการพิจารณาของทั้งสองกราฟแล้วนั้นไม่เกิดความเสถียรโมเดล LSTM แบบพื้นฐานที่ใช้ชุดข้อมูลการเสียชีวิตสะสมในการฝึกฝนโมเดลถึงแม่ว่ากราฟทั้งสองนั้นจะมีแนวโน้มการบรรจบกันของเส้นกราฟก็ตาม



ภาพ 70 ตัวอย่าง Learning Curve ของโมเดล LSTM แบบพื้นฐานโดยใช้ชุดข้อมูลการเสียชีวิตสะสมของจังหวัดบึงกาฬ

จากภาพ 70 แนวโน้มของเส้นกราฟ Training Loss นั้นมีแนวโน้มของการฝึกฝนโมเดลที่ลดลงและเริ่มมีความคงที่แต่ยังไม่นับว่าเข้าสู่สถานะเสถียรโดยสมบูรณ์ เมื่อพิจารณาเส้นกราฟของ Validation Loss แล้วนั้นพบว่าแนวโน้มของเส้นกราฟไม่มีความเสถียรเป็นอย่างมากเนื่องจากมีจุดยอดของกราฟที่สูงมาก และเมื่อพิจารณาทั้งสองเส้นกราฟแล้วพบว่าในช่วงแรกทั้งสองกราฟมีแนวโน้มที่บรรจบกันแต่เมื่อมีการฝึกฝนในรอบที่มากขึ้นพบว่ากราฟทั้งสองมีแนวโน้มที่เริ่มห่างออกจากกันจนกระทั่งไม่บรรจบกันในที่สุด

4.1.2 ผลลัพธ์การทำนายโดยการใช้โมเดล LSTM แบบพื้นฐาน

ในการพัฒนาโมเดลในหัวข้อนี้ ได้ทำการทดลองและวิเคราะห์การทำนายสถานการณ์การแพร่ระบาดของโควิด-19 โดยใช้โมเดล LSTM แบบพื้นฐาน ซึ่งเป็นเทคนิคการเรียนรู้เชิงลึกที่มีความสามารถในการจัดการกับข้อมูลอนุกรมเวลา โมเดล LSTM นี้นำมาใช้เพื่อทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสมจากข้อมูลที่ได้จากกรมควบคุมโรคของประเทศไทย ในการทดสอบประสิทธิภาพของโมเดลได้ใช้ชุดข้อมูลที่มีการรวบรวมข้อมูลในแต่ละวันจากทั้ง 77 จังหวัดของประเทศไทย ผลลัพธ์ที่ได้จากการทำนายด้วยโมเดล LSTM แบบพื้นฐานนี้จะยกตัวอย่างสองลำดับจังหวัดที่มีความแม่นยำในการทำนายที่สุดสองจังหวัดในการทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสม

การวิเคราะห์ข้อมูลแบบ Time series เป็นวิธีการที่ใช้ในการวิเคราะห์ชุดข้อมูลที่เก็บสะสมตามช่วงเวลา ข้อมูลในอนุกรมเวลาได้บันทึกเป็นจุดข้อมูลที่มีระยะเวลาคงที่ตลอดช่วงเวลาที่กำหนดไว้

ไม่ใช่การบันทึกข้อมูลเมื่อมีเหตุการณ์ที่เกิดขึ้นหรือสุ่มมาเพียงครั้งเดียวแต่จะเป็นการบันทึกข้อมูลในช่วงเวลาที่ต่อเนื่องกัน สิ่งที่ทำให้ข้อมูลอนุกรมเวลาแตกต่างจากข้อมูลอื่นๆคือการวิเคราะห์สามารถแสดงให้เห็นถึงวิธีการเปลี่ยนแปลงของตัวแปรตามเวลา กล่าวคือ ตัวแปรเวลาเป็นตัวแปรที่สำคัญเพื่อแสดงให้เห็นถึงข้อมูลเปลี่ยนแปลงที่จุดข้อมูลต่างๆและผลลัพธ์สุดท้าย การวิเคราะห์อนุกรมเวลาจำเป็นต้องมีจำนวนข้อมูลที่มากเพียงพอเพื่อให้มั่นใจในความเสถียรและความน่าเชื่อถือของผลการการวิเคราะห์อนุกรมเวลายังสามารถนำมาใช้ในการทำนายโดยการคาดคะเนข้อมูลในอนาคตจากข้อมูลที่มีในอดีต

การใช้ตัวชี้วัดประสิทธิภาพของโมเดลประเภทอนุกรมเวลานั้นจำเป็นต้องใช้เมตริกเฉพาะของการวัดประสิทธิภาพเนื่องจากการวัดประสิทธิภาพของโมเดลแบบอนุกรมเวลานั้นจะแตกต่างกับการทำ Classification ดั้งนั้นจึงจำเป็นต้องเลือกเครื่องมือที่ถูกต้องสำหรับการวัดประสิทธิภาพของโมเดลโดยเมตริกการประเมินของชุดข้อมูลแนวเวลา (Time Series Evaluation Metrics)[133, 134] เป็นเครื่องมือที่ใช้ในการประเมินประสิทธิภาพและความแม่นยำของโมเดลทำนายแนวเวลา ข้อมูลที่ใช้ในการประเมินเป็นชุดข้อมูลที่เก็บค่าต่างๆ ในช่วงเวลาที่กำหนด เมตริกเหล่านี้ช่วยในการวัดว่าการทำนายของโมเดลมีความสอดคล้องกับค่าจริงในช่วงเวลานั้นเพียงใด โดยในงานวิจัยนี้ได้นำเสนอทั้งหมด 3 ตัวชี้วัดเพื่อทำการวัดประสิทธิภาพของโมเดล ได้แก่

ก) Mean Squared Error (MSE)

Mean Squared Error คือค่าเฉลี่ยของค่าผลต่างกำลังสองที่มีความสัมพันธ์เชิงบวกระหว่างค่าจริงและค่าที่โมเดลทำนาย ค่า MSE ที่น้อยจะแสดงถึงความแม่นยำของโมเดลทำนาย ซึ่งหมายความว่าค่าการแตกต่างระหว่างค่าจริงและค่าทำนายของโมเดลในแต่ละจุดข้อมูลจะมีน้อยที่สุด โดย MSE เป็นตัวบ่งชี้ที่ใช้วัดคุณภาพของโมเดลทำนายหรือตัวทำนาย นอกจากนี้ MSE สามารถพิจารณาความแปรปรวน (ความแตกต่างระหว่างค่าที่คาดการณ์ไว้กับค่าจริง) และความห่างของค่าทำนายจากค่าจริง (ความคลาดเคลื่อนของค่าทำนายจากค่าจริง) โดยสมการของ MSE นั้นอธิบายไว้ในสมการที่ (12)

$$MSE(y, y) = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i)^2 \tag{12}$$

โดยที่ n คือ จำนวนตัวอย่างในชุดข้อมูล

y คือ ค่าจริง

y คือ ค่าที่โมเดลทำนาย

ข) Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) [135] เป็นตัววัดที่ใช้ในการวัดความแม่นยำของโมเดลในการทำนาย time series ที่นิยมใช้มากที่สุด การคำนวณ RMSE เป็นการหาค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนในรูปของค่าเต็มหรือความแตกต่างระหว่างค่าจริงและค่าทำนายของโมเดล การรายงาน RMSE มักจะเป็นหน่วย "ความผิดพลาดต่อหน่วย" ซึ่งช่วยให้คุณสามารถเปรียบเทียบโมเดลที่ต่างกันได้ในระดับเท่ากัน

การคำนวณ RMSE จะเริ่มต้นด้วยการคำนวณค่าความคลาดเคลื่อนสำหรับแต่ละจุดข้อมูล ซึ่งสามารถทำได้โดยหาผลต่างระหว่างค่าจริงและค่าที่โมเดลทำนาย และต้องยกกำลังสองของค่าผลต่างเหล่านี้และหาค่าเฉลี่ย ผลลัพธ์ที่ได้จากขั้นตอนนี้เรียกว่า Mean Square Error (MSE) สุดท้ายจะต้องหาค่ารากที่สองของค่าเฉลี่ยเพื่อให้ได้ค่า RMSE ดังสมกานที่ (13)

$$RMSE(y, y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2} \tag{13}$$

โดยที่ n คือ จำนวนตัวอย่างในชุดข้อมูล

y คือ ค่าจริง

y คือ ค่าที่โมเดลทำนาย

ค) Mean Absolute Error (MAE)

Mean Absolute Error (MAE) เป็นตัวบ่งชี้ที่ใช้วัดความแม่นยำของโมเดลทำนายแนวเวลาในการทำนายชุดข้อมูลสำหรับแต่ละจุดข้อมูล [136] การคำนวณ MAE คือการหาค่าเฉลี่ยของค่าผลต่างที่มีความสัมพันธ์เชิงบวกระหว่างค่าจริงและค่าที่โมเดลทำนาย สำหรับค่า MAE ไม่มีคำตอบแน่ชัดเกี่ยวกับค่าที่เหมาะสมสำหรับโมเดลทำนายแนวเวลา แต่กฎบังคับที่ดีคือค่า MAE ควรจะเป็นต่ำเท่าที่เป็นไปได้ พร้อมกับความแม่นยำที่สูง สมการของ MAE แสดงในสมการที่ (14)

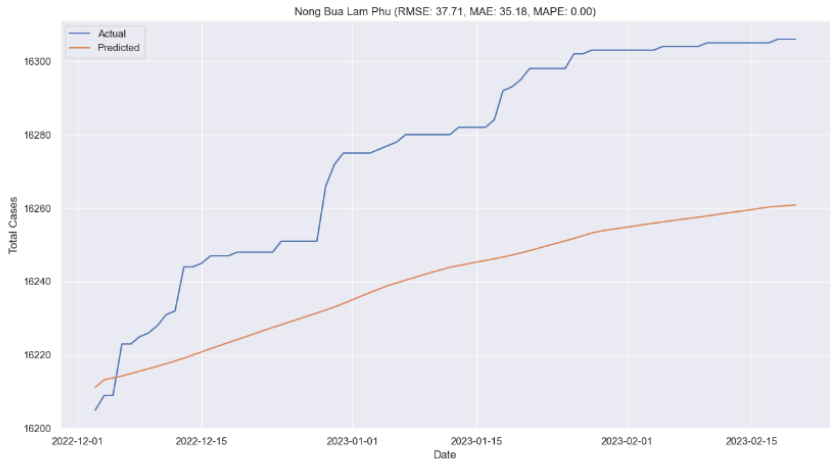
$$MAE(y, y) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i\right| \tag{14}$$

โดยที่ n คือ จำนวนตัวอย่างในชุดข้อมูล

y คือ ค่าจริง

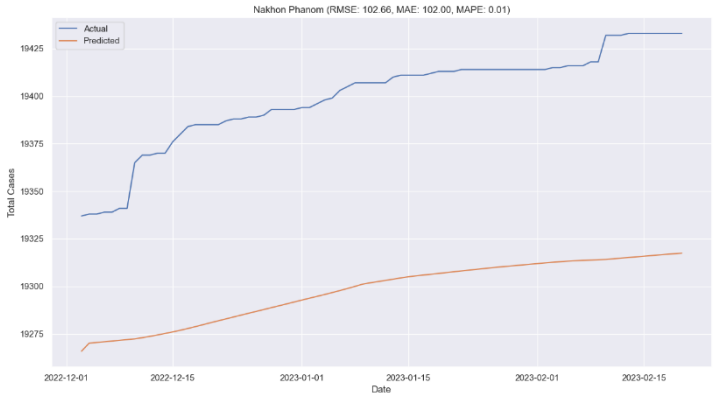
y คือ ค่าที่โมเดลทำนาย

ตัวอย่างผลการทำนายของโมเดล LSTM แบบพื้นฐาน โดยผลการทำนายของโมเดล LSTM แบบพื้นฐานนั้นได้แสดงสองจังหวัดที่มีความแม่นยำมากที่สุดของชุดข้อมูลการติดเชื้อสะสมและการเสียชีวิตสะสม ผลการทำนายนั้นเป็นการเปรียบเทียบค่าจริงและค่าของการทำนายผ่านเมตริกซ์การวัด คือ MAE MSE RMSE



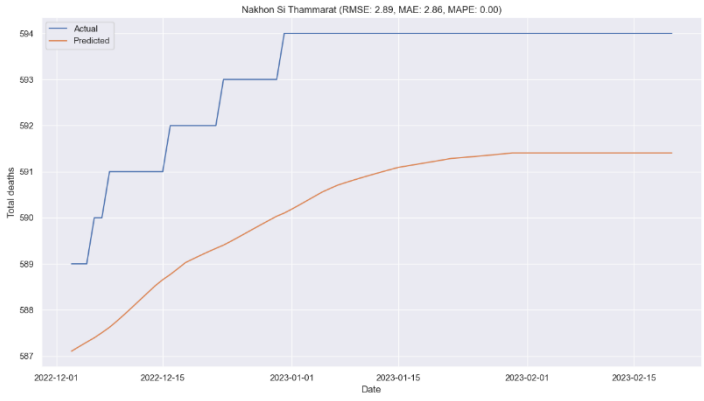
ภาพ 71 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM แบบพื้นฐาน ของจังหวัดหนองบัวลำภู

ภาพ 71 แสดงกราฟของจังหวัดหนองบัวลำภูซึ่งเป็นจังหวัดที่มีความแม่นยำมากที่สุดของโมเดล LSTM แบบพื้นฐาน แสดงจำนวนผู้ติดเชื้อสะสม (Actual) และจำนวนผู้ติดเชื้อสะสมที่ทำนาย (Predicted) โดยเส้นสีน้ำเงินแทนข้อมูลจริงและเส้นสีส้มแทนข้อมูลที่ทำนายจากกราฟจะเห็นว่าจำนวนผู้ติดเชื้อสะสมที่คาดการณ์มีความใกล้เคียงกับข้อมูลจริงในช่วงเวลาต่าง ๆ ค่าตัวชี้วัดการทำนาย RMSE คือ 37.71 และ MAE คือ 35.18 แม้ค่าความคลาดเคลื่อนเฉลี่ย (MAE) จะดูสูง แต่เมื่อพิจารณาจากหน่วยแกน Y ซึ่งมีค่าระหว่าง 16200 ถึง 16300 ค่าความคลาดเคลื่อนนี้ถือว่ายอมรับได้ เส้นแนวโน้มของกราฟแสดงถึงการเพิ่มขึ้นของจำนวนผู้ติดเชื้อสะสมอย่างต่อเนื่อง แม้ว่าจะมีช่วงเวลาที่ข้อมูลจริงมีความผันผวน แต่เส้นคาดการณ์แสดงให้เห็นถึงการเพิ่มขึ้นอย่างมีแนวโน้มที่สม่ำเสมอ



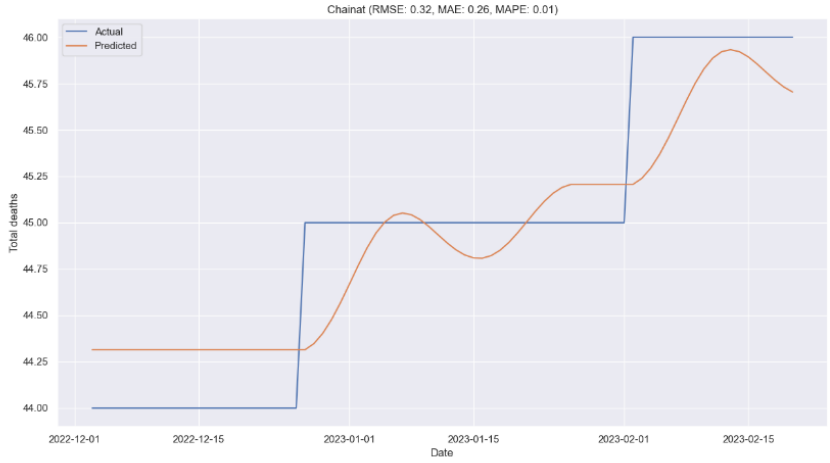
ภาพ 72 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM แบบพื้นฐาน ของจังหวัดนครปฐม

ภาพ 72 แสดงกราฟของจังหวัดนครปฐมซึ่งเป็นจังหวัดที่มีความแม่นยำเป็นลำดับที่ 2 รองจากจังหวัดหนองบัวลำภู ซึ่งแสดงจำนวนผู้ติดเชื้อสะสมและจำนวนผู้ติดเชื้อสะสมที่ทำนายจากกราฟจะเห็นว่าจำนวนผู้ติดเชื้อสะสมที่คาดการณ์มีความใกล้เคียงกับข้อมูลจริงในช่วงเวลาต่าง ๆ ค่าตัวชี้วัดการทำนาย RMSE คือ 102.66 และ MAE คือ 102.00 ค่า MAE แสดงถึงความคลาดเคลื่อนเฉลี่ยที่อยู่ในระดับที่ค่อนข้างสูง แต่เมื่อพิจารณาจากหน่วยแกน Y ซึ่งมีค่าระหว่าง 19200 ถึง 19500 ค่าความคลาดเคลื่อนนี้ยังอยู่ในระดับที่ยอมรับได้ เส้นแนวโน้มของกราฟแสดงถึงการเพิ่มขึ้นของจำนวนผู้ติดเชื้อสะสมอย่างต่อเนื่อง แม้ว่าจะมีความผันผวนในข้อมูลจริงบางช่วงเวลา



ภาพ 73 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM แบบพื้นฐาน ของจังหวัดนครศรีธรรมราช

ภาพ 73 แสดงกราฟของจังหวัดนครศรีธรรมราช ซึ่งเป็นจังหวัดที่มีความแม่นยำมากที่สุดของโมเดล LSTM แบบพื้นฐานในการทำนายจำนวนผู้ติดเชื้อสะสม โดยภาพ 73 แสดงจำนวนผู้เสียชีวิตสะสมและจำนวนผู้เสียชีวิตสะสมที่ทำนายโดยเส้นสีน้ำเงินแทนข้อมูลจริงและเส้นสีส้มแทนข้อมูลที่ทำนายค่าตัวชี้วัดการทำนาย RMSE คือ 2.89 และ MAE คือ 2.86 จากกราฟจะเห็นว่าจำนวนผู้เสียชีวิตสะสมที่คาดการณ์มีความใกล้เคียงกับข้อมูลจริงในช่วงเวลาต่าง ๆ เส้นแนวโน้มของกราฟแสดงถึงการเพิ่มขึ้นของจำนวนผู้เสียชีวิตสะสม โดยเส้นสีส้มที่แสดงการคาดการณ์มีแนวโน้มที่สม่ำเสมอและเพิ่มขึ้นอย่างต่อเนื่อง แม้ว่าจะมีช่วงเวลาที่ข้อมูลจริงมีการเพิ่มขึ้นแบบเป็นขั้น ๆ แต่การคาดการณ์ยังคงแสดงถึงแนวโน้มที่ถูกต้องโดยรวม



ภาพ 74 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM แบบพื้นฐาน ของจังหวัดชัยนาท

ภาพ 74 แสดงกราฟของจังหวัดชัยนาทซึ่งเป็นจังหวัดที่มีความแม่นยำเป็นลำดับที่ 2 รองจากจังหวัดนครศรีธรรมราช ค่าตัวชี้วัดการทำนาย RMSE คือ 0.32 และ MAE คือ 0.26 เส้นแนวโน้มของกราฟแสดงถึงการเพิ่มขึ้นของจำนวนผู้เสียชีวิตสะสมอย่างเป็นแบบต่อเนื่อง ข้อมูลจริงมีการเพิ่มขึ้นและในขณะเดียวกันเส้นคาดการณ์แสดงถึงการคาดการณ์ที่มีความผันผวนเล็กน้อยแต่ยังคงสอดคล้องกับแนวโน้มของข้อมูลจริง การคาดการณ์นี้มีความแม่นยำสูง ซึ่งแสดงให้เห็นว่าโมเดลสามารถทำนายการเปลี่ยนแปลงของจำนวนผู้เสียชีวิตสะสมในจังหวัดชัยนาทได้อย่างมีประสิทธิภาพ

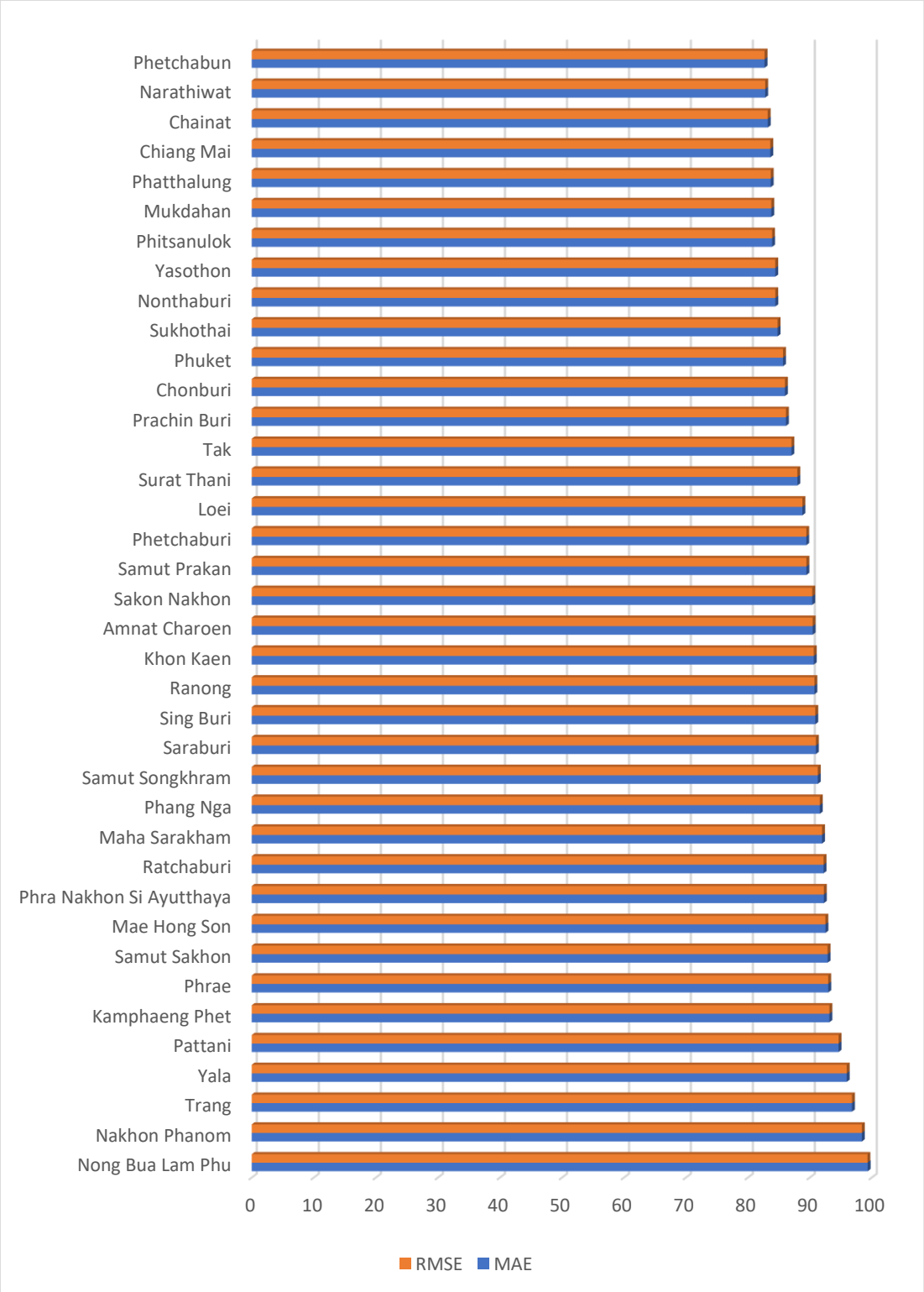
4.1.3 ผลการทดสอบประสิทธิภาพของโมเดล LSTM แบบพื้นฐาน

เพื่อประเมินประสิทธิภาพของโมเดล LSTM แบบพื้นฐานในการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 จึงได้ทำการทดสอบและวิเคราะห์ผลลัพธ์ที่ได้จากโมเดล โดยใช้ชุดข้อมูลการติดเชื้อสะสมและการเสียชีวิตสะสมของแต่ละจังหวัดในประเทศไทย ในการรายงานผลการทดลองนี้ ผู้วิจัยได้นำเสนอข้อมูลในรูปแบบของกราฟและตาราง ซึ่งประกอบด้วยภาพ 75 ตาราง 7 และภาพ 76 ตาราง 8 โดยตาราง 7 และ 8 คอลัมน์ Province คือชื่อจังหวัด คอลัมน์ ACC (MAE) คือค่าความถูกต้องแม่นยำที่คำนวณโดยใช้ค่า MAE คอลัมน์ ACC (RMSE) คือค่าความถูกต้องแม่นยำที่คำนวณโดยใช้ค่า RMSE การแปลงเป็นร้อยละความถูกต้องและเรียงลำดับจากความแม่นยำมากสุดไปน้อยที่สุดเพื่อให้เห็นถึงการเปรียบเทียบและการวิเคราะห์ผลลัพธ์ของโมเดลอย่างละเอียด ตารางผลลัพธ์ที่ได้จากการวัดค่า MAE RMSE MSE ของการติดเชื้อสะสมและการเสียชีวิตสะสมของโมเดล LSTM แบบพื้นฐานนั้นได้แสดงในภาคผนวก โดยจังหวัดที่มีความถูกต้องมากที่สุดในการทำนายชุดข้อมูลการติดเชื้อสะสมของโมเดล LSTM แบบพื้นฐานได้แก่ จังหวัดหนองบัวลำภูมีความถูกต้องร้อยละ 99.37 จังหวัดนครพนมมีความถูกต้องร้อยละ 98.42 จังหวัดตรังมีความถูกต้องร้อยละ 96.82 จังหวัดยะลามีความถูกต้องร้อยละ 95.98 และจังหวัดปัตตานีมีความถูกต้องร้อยละ 94.66 ซึ่งผลการทำนายของ 2 อันดับที่มีความแม่นยำมากที่จะถูกนำไปวิเคราะห์ในขั้นตอนถัดไป

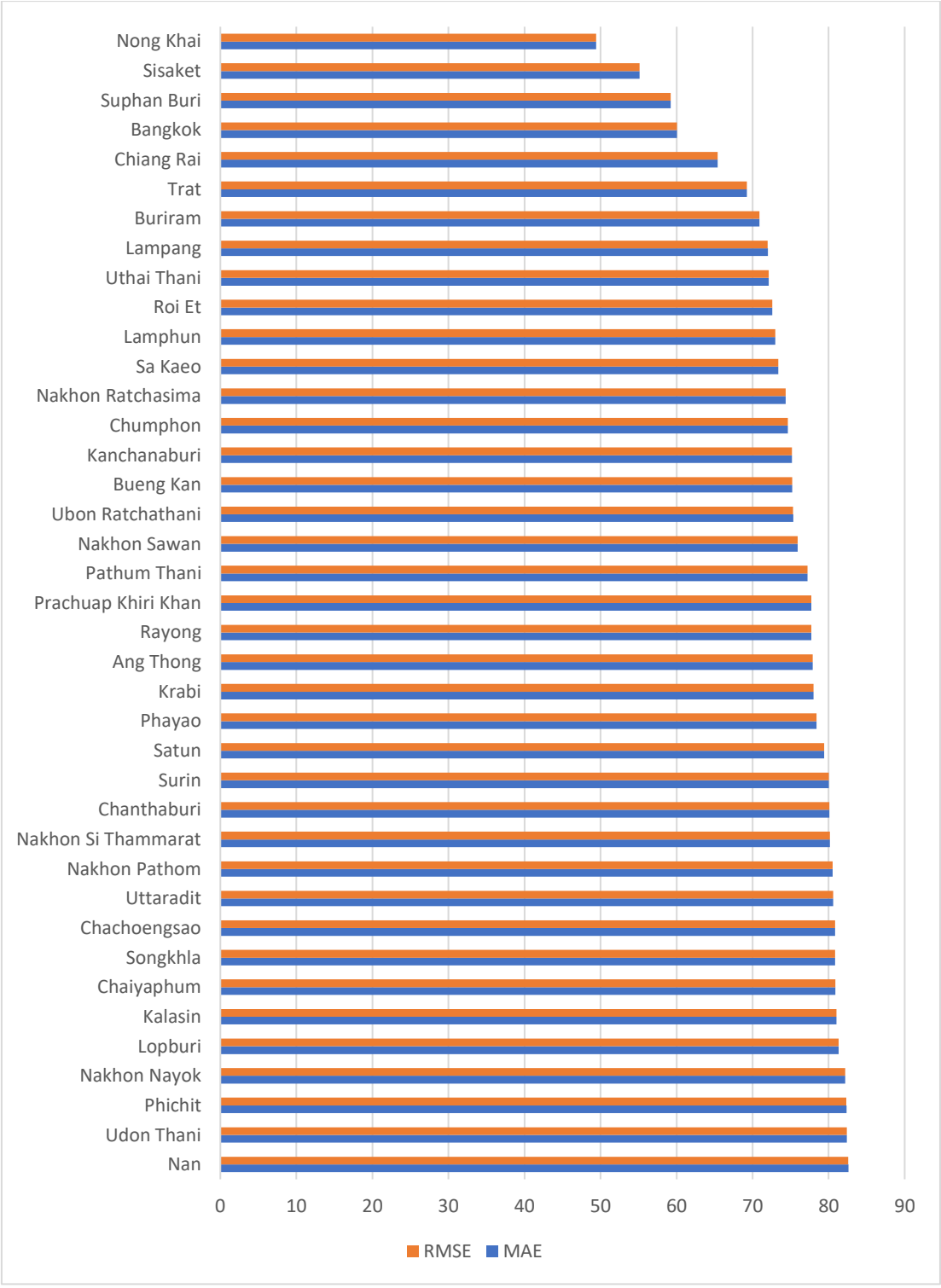
จังหวัดที่มีความแม่นยำน้อยที่สุดในการทำนายของโมเดล LSTM พื้นฐานในการทำนายชุดข้อมูลการติดเชื้อสะสมได้แก่ จังหวัดเชียงรายมีความถูกต้องร้อยละ 65.39 กรุงเทพมหานครมีความถูกต้องร้อยละ 60.04 จังหวัดสุพรรณบุรีมีความถูกต้องร้อยละ 59.21 จังหวัดศรีสะเกษมีความถูกต้องร้อยละ 55.13 และจังหวัดหนองคายมีความถูกต้องร้อยละ 49.43 ดังแสดงในภาพ 75 ตาราง 7 โดยสองจังหวัดที่มีความแม่นยำมากที่สุดของโมเดลนี้จะถูกนำไปวิเคราะห์ในขั้นตอนถัดไปเช่นกัน

จังหวัดที่มีความแม่นยำมากที่สุดในการทำนายของโมเดล LSTM พื้นฐานในการทำนายชุดข้อมูลการเสียชีวิตสะสมได้แก่ จังหวัดนครศรีธรรมราชมีความถูกต้องร้อยละ 98.65 จังหวัดชัยนาทมีความถูกต้องร้อยละ 98.65 อุดรธานีมีความถูกต้องร้อยละ 98.42 จังหวัดพัทลุงมีความถูกต้องร้อยละ 98.42 และจังหวัดพิจิตรมีความถูกต้องร้อยละ 98.30 ดังแสดงในภาพ 76 ตาราง 8 โดยสองจังหวัดที่มีความแม่นยำมากที่สุดของโมเดลนี้จะถูกนำไปวิเคราะห์ในขั้นตอนถัดไปเช่นกัน

จังหวัดที่มีความแม่นยำต่ำที่สุดในการทำนายของโมเดล LSTM พื้นฐานในการทำนายชุดข้อมูลการเสียชีวิตสะสมได้แก่ จังหวัดพะเยามีความถูกต้องร้อยละ -37.80 จังหวัดอุทัยธานีมีความถูกต้องร้อยละ -90.54 จังหวัดขอนแก่นมีความถูกต้องร้อยละ -90.84 จังหวัดนครราชสีมามีความถูกต้องร้อยละ -120.06 และจังหวัดเชียงใหม่มีความถูกต้องร้อยละ -185.83 ดังแสดงในภาพ 76 ตาราง 8 โดยสองจังหวัดที่มีความแม่นยำต่ำที่สุดของโมเดลนี้จะถูกนำไปวิเคราะห์ในขั้นตอนถัดไปเช่นกัน



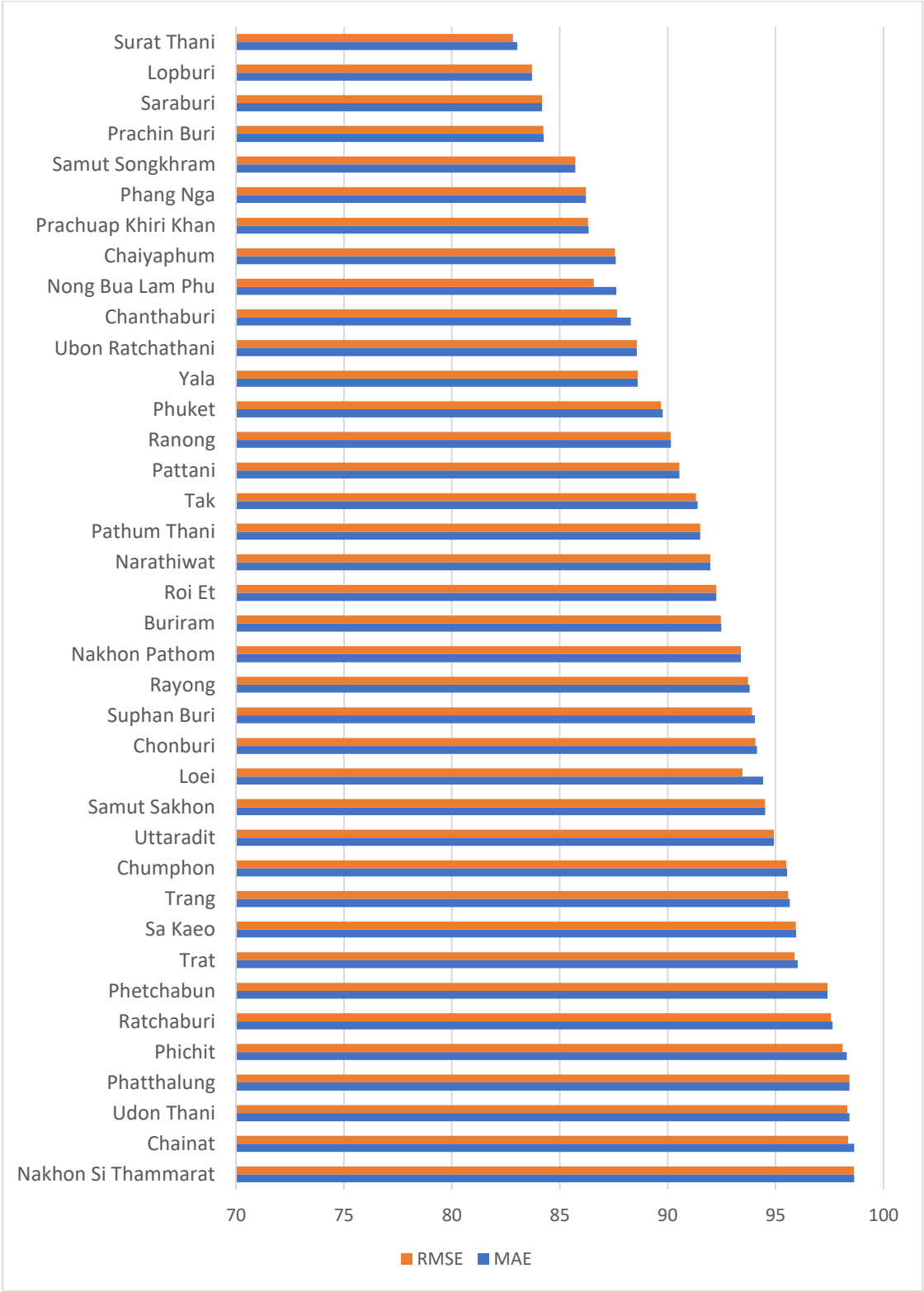
ภาพ 75 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM พื้นฐานของทุกจังหวัด



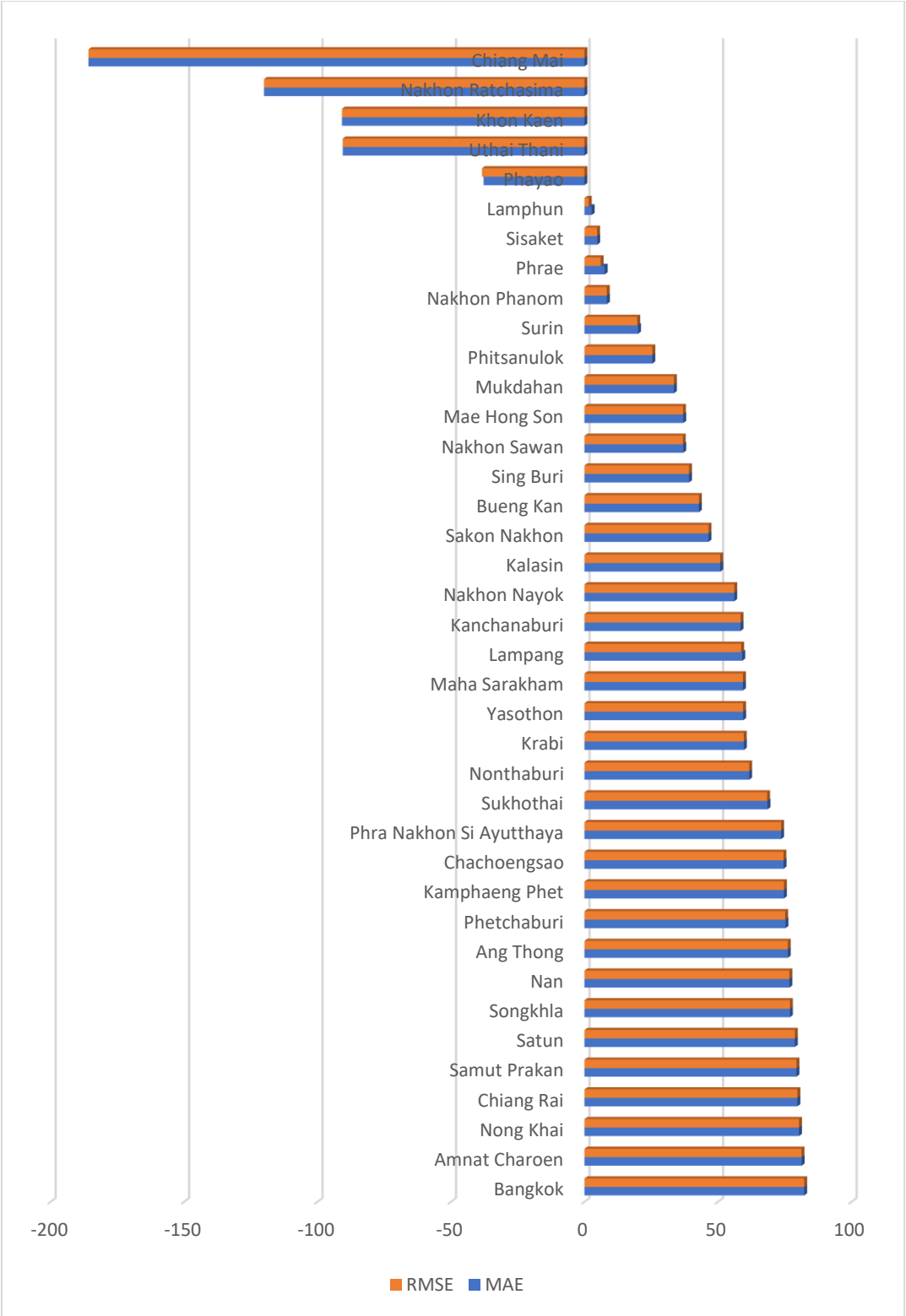
ภาพ 75 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM พื้นฐานของทุกจังหวัด (ต่อ)

ตาราง 7 ผลการทำนายของโมเดล LSTM แบบพื้นฐานจากชุดมูลการติดเชื้อสะสม

Province	ACC (MAE)	ACC (RMSE)	Province	ACC (MAE)	ACC (RMSE)
Nong Bua Lam Phu	99.37	99.32	Nan	82.61	82.56
Nakhon Phanom	98.42	98.41	Udon Thani	82.39	82.39
Trang	96.82	96.82	Phichit	82.34	82.34
Yala	95.98	95.98	Nakhon Nayok	82.18	82.18
Pattani	94.66	94.66	Lopburi	81.32	81.31
Kamphaeng Phet	93.18	93.18	Kalasin	81.02	81.02
Phrae	92.99	92.98	Chaiyaphum	80.90	80.90
Samut Sakhon	92.89	92.89	Songkhla	80.86	80.86
Mae Hong Son	92.54	92.52	Chachoengsao	80.83	80.83
Phra Nakhon Si Ayutthaya	92.29	92.29	Uttaradit	80.59	80.59
Ratchaburi	92.23	92.23	Nakhon Pathom	80.52	80.52
Maha Sarakham	91.99	91.99	Nakhon Si Thammarat	80.18	80.18
Phang Nga	91.64	91.64	Chanthaburi	80.10	80.10
Samut Songkhram	91.31	91.31	Surin	80.03	80.02
Saraburi	91.00	91.00	Satun	79.40	79.40
Sing Buri	90.90	90.90	Phayao	78.42	78.42
Ranong	90.76	90.76	Krabi	78.00	78.00
Khon Kaen	90.66	90.66	Ang Thong	77.89	77.89
Amnat Charoen	90.47	90.47	Rayong	77.71	77.71
Sakon Nakhon	90.41	90.41	Prachuap Khiri Khan	77.71	77.70
Samut Prakan	89.46	89.46	Pathum Thani	77.22	77.22
Phetchaburi	89.44	89.44	Nakhon Sawan	75.92	75.92
Loei	88.82	88.81	Ubon Ratchathani	75.34	75.33
Surat Thani	88.01	88.01	Bueng Kan	75.21	75.21
Tak	87.05	87.05	Kanchanaburi	75.18	75.18
Prachin Buri	86.17	86.17	Chumphon	74.62	74.62
Chonburi	85.99	85.99	Nakhon Ratchasima	74.34	74.34
Phuket	85.69	85.69	Sa Kaeo	73.39	73.39
Sukhothai	84.82	84.82	Lamphun	72.99	72.98
Nonthaburi	84.45	84.45	Roi Et	72.60	72.60
Yasothon	84.45	84.45	Uthai Thani	72.13	72.13
Phitsanulok	83.93	83.93	Lampang	72.00	71.99
Mukdahan	83.81	83.81	Buriram	70.89	70.88
Phatthalung	83.71	83.71	Trat	69.24	69.23
Chiang Mai	83.64	83.64	Chiang Rai	65.39	65.39
Chainat	83.23	83.22	Bangkok	60.04	60.04
Narathiwat	82.81	82.81	Suphan Buri	59.21	59.21
Phetchabun	82.72	82.72	Sisaket	55.13	55.13
			Nong Khai	49.43	49.42



ภาพ 76 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM พื้นฐานของทุกจังหวัด



ภาพ 76 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM พื้นฐานของทุกจังหวัด (ต่อ)

ตาราง 8 ผลการทำนายของโมเดล LSTM แบบพื้นฐานจากชุดมูลการเสียชีวิตสะสม

Province	ACC (MAE)	ACC (RMSE)	Province	ACC (MAE)	ACC (RMSE)
Nakhon Si Thammarat	98.65	98.63	Bangkok	82.28	82.28
Chainat	98.65	98.37	Amnat Charoen	81.33	81.28
Udon Thani	98.42	98.33	Nong Khai	80.35	80.35
Phatthalung	98.42	98.42	Chiang Rai	79.73	79.73
Phichit	98.30	98.10	Samut Prakan	79.40	79.40
Ratchaburi	97.64	97.57	Satun	78.79	78.79
Phetchabun	97.41	97.41	Songkhla	76.96	76.96
Trat	96.02	95.88	Nan	76.75	76.75
Sa Kaeo	95.95	95.94	Ang Thong	76.15	76.13
Trang	95.66	95.58	Phetchaburi	75.28	75.16
Chumphon	95.53	95.50	Kamphaeng Phet	74.70	74.69
Uttaradit	94.93	94.93	Chachoengsao	74.57	74.53
Samut Sakhon	94.51	94.51	Phra Nakhon Si Ayutthaya	73.65	73.64
Loei	94.42	93.47	Sukhothai	68.54	68.40
Chonburi	94.14	94.07	Nonthaburi	61.70	61.69
Suphan Buri	94.04	93.91	Krabi	59.72	59.67
Rayong	93.80	93.72	Yasothon	59.44	59.40
Nakhon Pathom	93.40	93.39	Maha Sarakham	59.35	59.35
Buriram	92.49	92.47	Lampang	59.10	58.68
Roi Et	92.25	92.25	Kanchanaburi	58.45	58.45
Narathiwat	91.97	91.97	Nakhon Nayok	56.10	56.10
Pathum Thani	91.51	91.51	Kalasin	50.86	50.81
Tak	91.39	91.32	Sakon Nakhon	46.45	46.45
Pattani	90.54	90.54	Bueng Kan	42.87	42.87
Ranong	90.15	90.14	Sing Buri	39.18	39.16
Phuket	89.77	89.68	Nakhon Sawan	37.02	36.84
Yala	88.61	88.61	Mae Hong Son	37.01	36.88
Ubon Ratchathani	88.58	88.58	Mukdahan	33.50	33.50
Chanthaburi	88.29	87.66	Phitsanulok	25.42	25.42
Nong Bua Lam Phu	87.62	86.57	Surin	20.08	19.78
Chaiyaphum	87.59	87.56	Nakhon Phanom	8.41	8.38
Prachuap Khiri Khan	86.33	86.32	Phrae	7.60	6.07
Phang Nga	86.21	86.21	Sisaket	4.79	4.73
Samut Songkhram	85.72	85.72	Lamphun	2.72	1.63
Prachin Buri	84.26	84.24	Phayao	-37.80	-38.37
Saraburi	84.18	84.18	Uthai Thani	-90.54	-90.55
Lopburi	83.72	83.72	Khon Kaen	-90.84	-90.85
Surat Thani	83.03	82.83	Nakhon Ratchasima	-120.06	-120.06
			Chiang Mai	-185.83	-185.83

ผลการทำนายของโมเดล LSTM แบบพื้นฐานที่ใช้สำหรับการทำนายชุดข้อมูลผู้ติดเชื้อสะสมและการเสียชีวิตสะสมนั้นได้ทำการสรุปตามตาราง 9 ซึ่งเป็นการหาค่าเฉลี่ยร้อยละความถูกต้องของทุกจังหวัดที่โมเดลใช้ในการทำนาย

ตาราง 9 ผลการทดลองโมเดล LSTM แบบพื้นฐาน

Models	Prediction	RMSE	MAE	ร้อยละความถูกต้อง	
				คำนวณจาก RMSE	คำนวณจาก MAE
LSTM แบบพื้นฐาน	total cases	4828.07	4827.80	82.13	82.13
LSTM แบบพื้นฐาน	total deaths	46.04	45.97	62.44	62.56

ตาราง 9 แสดงผลการทดลองของโมเดล LSTM แบบพื้นฐานในการทำนายจำนวนผู้ติดเชื้อสะสมและผู้เสียชีวิตสะสมซึ่งสรุปจากตาราง 4 โดยใช้ตัวชี้วัด RMSE และ MAE เพื่อประเมินประสิทธิภาพของโมเดล ผลการทดลองแสดงค่าความคลาดเคลื่อนในการทำนายของโมเดลในการพยากรณ์ข้อมูลอนุกรมเวลาของจำนวนผู้ติดเชื้อสะสมและผู้เสียชีวิตสะสมในประเทศไทย ค่า RMSE และ MAE ของการทำนายจำนวนผู้ติดเชื้อสะสมอยู่ที่ 4828.07 และ 4827.80 ตามลำดับ โดยมีร้อยละความถูกต้องเท่ากับ 82.13 และ 82.13 ตามลำดับ ส่วนการทำนายจำนวนผู้เสียชีวิตสะสมมีค่า RMSE และ MAE อยู่ที่ 46.04 และ 45.97 ตามลำดับ โดยมีร้อยละความถูกต้องเท่ากับ 62.44 และ 62.56 ตามลำดับ

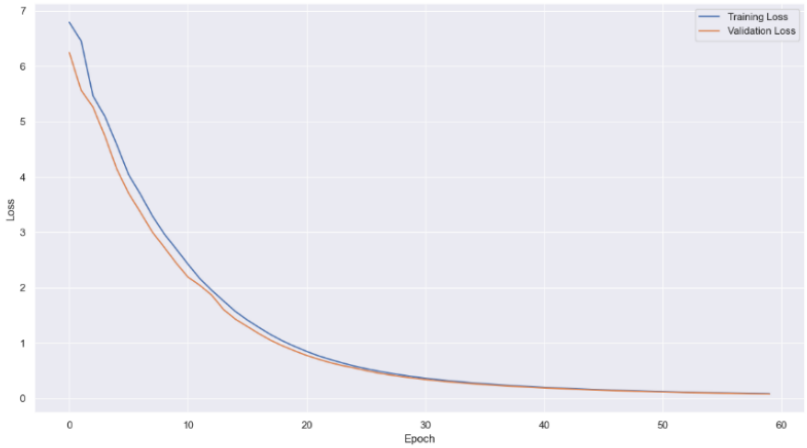
4.2 ผลการพัฒนาโมเดล LSTM + MLP

การพัฒนาโมเดลที่ผสมผสานระหว่าง LSTM และ MLP นั้นมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพในการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 ในประเทศไทย โดยโมเดล LSTM มีความสามารถในการจัดการข้อมูลอนุกรมเวลา ซึ่งช่วยในการจดจำแนวโน้มและรูปแบบของข้อมูลในอดีต ในขณะที่โมเดล MLP มีความสามารถในการจัดการกับข้อมูลสถิติคงที่ เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำและมีประสิทธิภาพมากยิ่งขึ้น ผลการทดสอบและการประเมินประสิทธิภาพของโมเดล LSTM + MLP นี้ ได้นำเสนอผ่านกราฟและตารางในส่วนต่าง ๆ ของการทดลอง เพื่อแสดงให้เห็นถึงความสามารถของโมเดลในการทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสมในแต่ละจังหวัดของประเทศไทยอย่างละเอียดและชัดเจน

4.2.1. การประเมินประสิทธิภาพของโมเดลในการฝึกอบรมโมเดล model validation

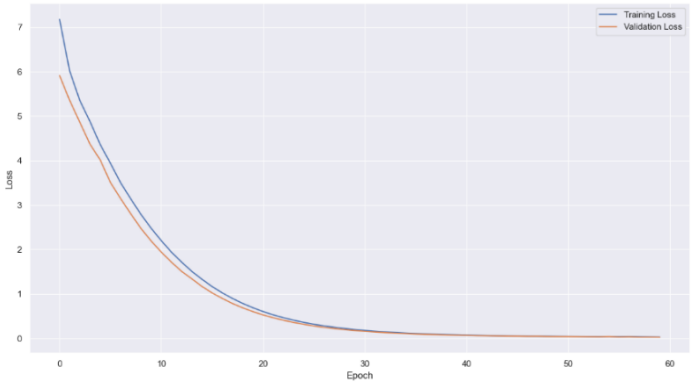
บรรยาย

การวิเคราะห์เพื่อประเมินประสิทธิภาพของโมเดล LSTM + MLP ได้ใช้การประเมินด้วย Learning Curve ในการพิจารณาแนวโน้มการลดลงของค่า Loss ของเส้น Validation loss และ Training loss เพื่อประเมินความเสถียรของโมเดล LSTM + MLP โดยใช้ชุดข้อมูลการติดเชื้อสะสมและการเสียชีวิตสะสม



ภาพ 77 ตัวอย่าง Learning Curve ของโมเดล LSTM + MLP โดยใช้ชุดข้อมูลการติดเชื้อสะสมของจังหวัดกรุงเทพมหานคร

จากภาพ 77 กราฟ Learning Curve ของโมเดล LSTM + MLP โดยใช้ชุดข้อมูลการติดเชื้อสะสมของจังหวัดกรุงเทพมหานคร แสดงให้เห็นว่าค่าความสูญเสีย (loss) ของทั้งชุดข้อมูลฝึก (Training Loss) และชุดข้อมูลทดสอบ (Validation Loss) ลดลงอย่างรวดเร็วในช่วงต้นของการฝึก 0-20 Epoch แสดงถึงการเรียนรู้และปรับปรุงโมเดลอย่างรวดเร็ว ช่วงกลางของการฝึก 20-40 Epoch ค่าความสูญเสียยังคงลดลงแต่ในอัตราที่ช้าลง ซึ่งบ่งบอกว่าโมเดลกำลังเข้าใกล้ค่าที่เหมาะสม หลังจาก 40 Epoch เป็นต้นไป ค่าความสูญเสียเริ่มคงที่ แสดงให้เห็นว่าโมเดลได้เข้าสู่ช่วงที่มีการปรับปรุงเล็กน้อยหรือไม่มีการปรับปรุงมากนัก โดยตลอดการฝึก ค่าความสูญเสียของชุดข้อมูลทดสอบมีความใกล้เคียงกับค่าความสูญเสียของชุดข้อมูลฝึก ซึ่งเป็นสิ่งที่บ่งชี้ว่าโมเดลไม่เกิดการ Overfitting

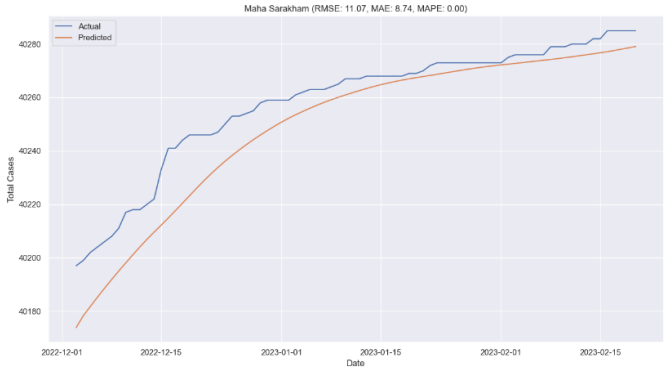


ภาพ 78 ตัวอย่าง Learning Curve ของโมเดล LSTM + MLP โดยใช้ชุดข้อมูลการเสียชีวิตสะสมของจังหวัดกรุงเทพมหานคร

จากภาพ 78 หากพิจารณาการวัดประสิทธิภาพของโมเดล LSTM + MLP โดยการใช้ชุดข้อมูลฝึกของการเสียชีวิตสะสมโดยการวิเคราะห์ด้วย Learning Curve ของทุกจังหวัดนั้นมีความใกล้เคียงกันมากคือ พบว่าลักษณะของกราฟ Training Loss และ Validation Loss มีความคล้ายคลึงกันกับกราฟของค่าฟังก์ชันการสูญเสียที่ได้จากการฝึกอบรมชุดข้อมูลการเสียชีวิตสะสมโดยที่กราฟทั้งสองเส้นนั้นบรรจบกันและมีความเสถียรที่คงที่ด้วยเช่นกันซึ่งมีรูปร่างเดียวกันและสอดคล้องกับการทำนายโดยใช้ข้อมูลการติดเชื้อสะสมด้วยเช่นกัน ซึ่งเป็นตัวบ่งชี้ว่าโมเดลมีความสเถียรมากพอในการทำนายข้อมูลทั้ง 2 รูปแบบ

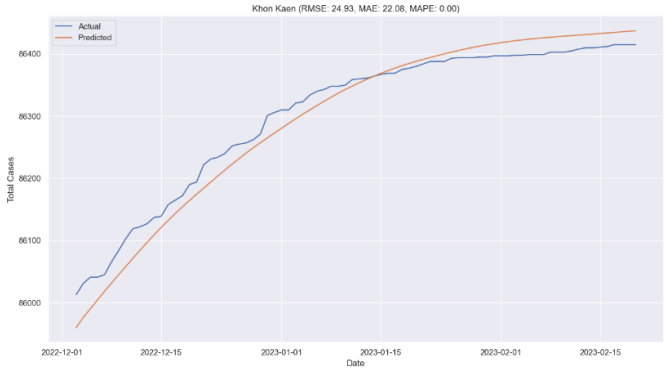
4.2.2 ผลลัพธ์การทำนายโดยการใช้โมเดล LSTM + MLP

เพื่อเพิ่มประสิทธิภาพและความแม่นยำในการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 ในประเทศไทย การพัฒนาโมเดลที่ผสมผสานระหว่าง LSTM และ MLP ได้นำมาใช้ การผสานกันระหว่าง LSTM ที่สามารถจัดการกับข้อมูลอนุกรมเวลาและ MLP ที่จัดการกับข้อมูลแบบคงที่ทำให้เกิดโมเดลที่มีความสามารถในการวิเคราะห์และทำนายแนวโน้มการแพร่ระบาดได้อย่างมีประสิทธิภาพ ในส่วนนี้จะนำเสนอผลลัพธ์ที่ได้จากการใช้โมเดล LSTM + MLP ในการทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสม โดยผลการทดสอบที่ได้จะแสดงให้เห็นถึงความแม่นยำและประสิทธิภาพของโมเดลในการทำนายดังภาพ 79



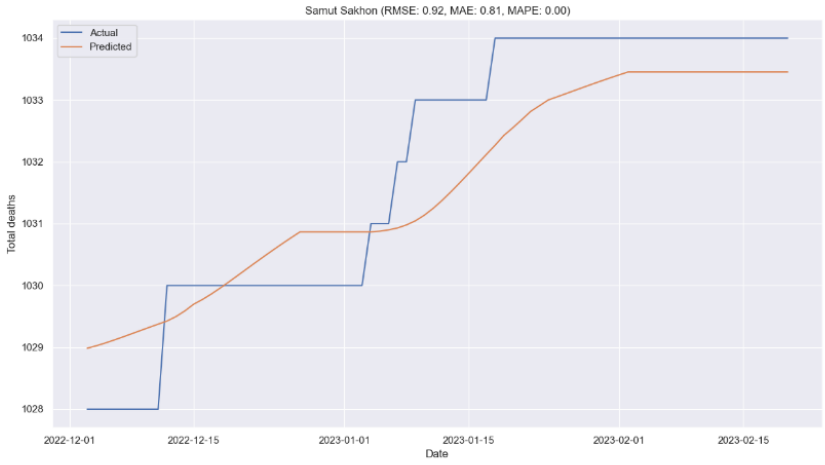
ภาพ 79 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM + MLP ของจังหวัดมหาสารคราม

ภาพ 79 คือกราฟของจังหวัดมหาสารครามโดยเป็นจังหวัดที่มีความถูกต้องมากที่สุดในการทำนายโดยใช้โมเดล LSTM + MLP ซึ่งผลการทำนายโมเดลการทำนายแสดงค่า RMSE ที่ 11.07 และ MAE ที่ 8.74 ซึ่งแสดงถึงความแม่นยำที่สูงกว่าเมื่อเปรียบเทียบกับขอนแก่น ค่า RMSE และ MAE ที่ต่ำกว่าแสดงว่าโมเดลนี้สามารถทำนายได้ใกล้เคียงกับค่าจริงมากขึ้น โดยเส้นทำนาย (สีส้ม) จะมีแนวโน้มใกล้เคียงกับเส้นค่าจริง (สีน้ำเงิน) มากขึ้น แสดงให้เห็นว่าค่าที่ทำนายมีความแม่นยำและสอดคล้องกับค่าจริงมากกว่า กราฟของมหาสารคามแสดงถึงค่าทำนายและค่าจริงมีการเคลื่อนไหวอย่างใกล้ชิด ทำให้เห็นภาพรวมของแนวโน้มการเติบโตในช่วงเวลาต่างๆ อย่างชัดเจน ความคลาดเคลื่อนระหว่างค่าจริงและค่าทำนายมีน้อยมากเมื่อเทียบกับข้อมูลจริงในแนวแกน y



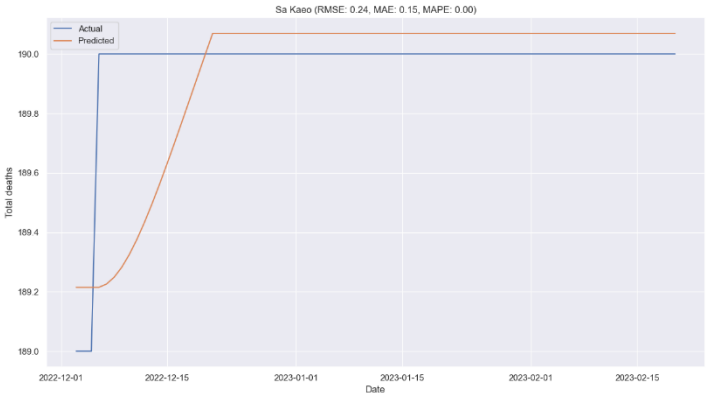
ภาพ 80 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM + MLP ของจังหวัดขอนแก่น

ในกรณีของจังหวัดขอนแก่นซึ่งเป็นจังหวัดที่มีความแม่นยำเป็นลำดับที่ 2 นั้นได้ผลลัพธ์ตามในภาพ 80 กราฟแสดงจำนวนผู้ติดเชื้อสะสมและจำนวนผู้ติดเชื้อสะสมที่คาดการณ์โดยมีเส้นสีน้ำเงินแทนข้อมูลจริงและเส้นสีส้มแทนข้อมูลทำนายค่าตัวชี้วัด RMSE คือ 24.93 และ MAE คือ 22.08 ซึ่งแสดงว่าโมเดลมีความสามารถในการคาดการณ์แนวโน้มได้อย่างมีประสิทธิภาพ แม้ว่าจะมีความคลาดเคลื่อนบ้าง เส้นสีส้มที่แสดงค่าคาดการณ์มีแนวโน้มสูงขึ้นเล็กน้อยเมื่อเทียบกับข้อมูลจริง ซึ่งบ่งบอกว่าโมเดลมีการคาดการณ์จำนวนผู้ติดเชื้อที่สูงกว่าค่าจริง การคาดการณ์นี้ยังคงมีความแม่นยำในระดับหนึ่ง



ภาพ 81 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM + MLP ของจังหวัดสมุทรสาคร

โมเดลการทำนายสำหรับจังหวัดสมุทรสาครในภาพ 81 แสดงค่า RMSE ที่ 0.92 และ MAE ที่ 0.81 กราฟแสดงค่าจริงและค่าทำนายในช่วงธันวาคม 2022 ถึงกุมภาพันธ์ 2023 ค่าจริงมีการเปลี่ยนแปลงเป็นขั้นๆ โดยค่าทำนายแสดงแนวโน้มที่ราบเรียบและเพิ่มขึ้นอย่างต่อเนื่อง โดยเฉพาะในช่วงที่ค่าจริงมีการเพิ่มขึ้นอย่างรวดเร็ว ค่าทำนายสามารถติดตามแนวโน้มนี้ได้อย่างดี แม้จะมีความคลาดเคลื่อนเล็กน้อยในบางจุด แต่ความคลาดเคลื่อนนี้ไม่ได้ส่งผลกระทบอย่างมีนัยสำคัญ เนื่องจากค่า MAE ต่ำ แสดงถึงความแม่นยำที่ดีของโมเดล



ภาพ 82 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM + MLP ของจังหวัดสระแก้ว

สำหรับจังหวัดสระแก้วในภาพ 82 คือโมเดลการทำนายแสดงค่า RMSE ที่ 0.24 และ MAE ที่ 0.15 กราฟแสดงค่าจริงและค่าทำนายในช่วงธันวาคม 2565 ถึงกุมภาพันธ์ 2566 ค่าจริงมีการเปลี่ยนแปลงอย่างรวดเร็วและคงที่ในช่วงเวลาที่เหลือ ค่าทำนายมีแนวโน้มใกล้เคียงกับค่าจริงอย่างมาก โดยเฉพาะในช่วงที่ค่าจริงคงที่ ค่าทำนายมีการเคลื่อนไหวในลักษณะเดียวกันและสามารถติดตามการเปลี่ยนแปลงได้อย่างแม่นยำ ความคลาดเคลื่อนระหว่างค่าจริงและค่าทำนายมีน้อยมาก ซึ่งบ่งชี้ว่าโมเดลมีความเสถียรและความน่าเชื่อถือสูงในการทำนาย

4.2.3 ผลการทดสอบประสิทธิภาพของโมเดล LSTM + MLP

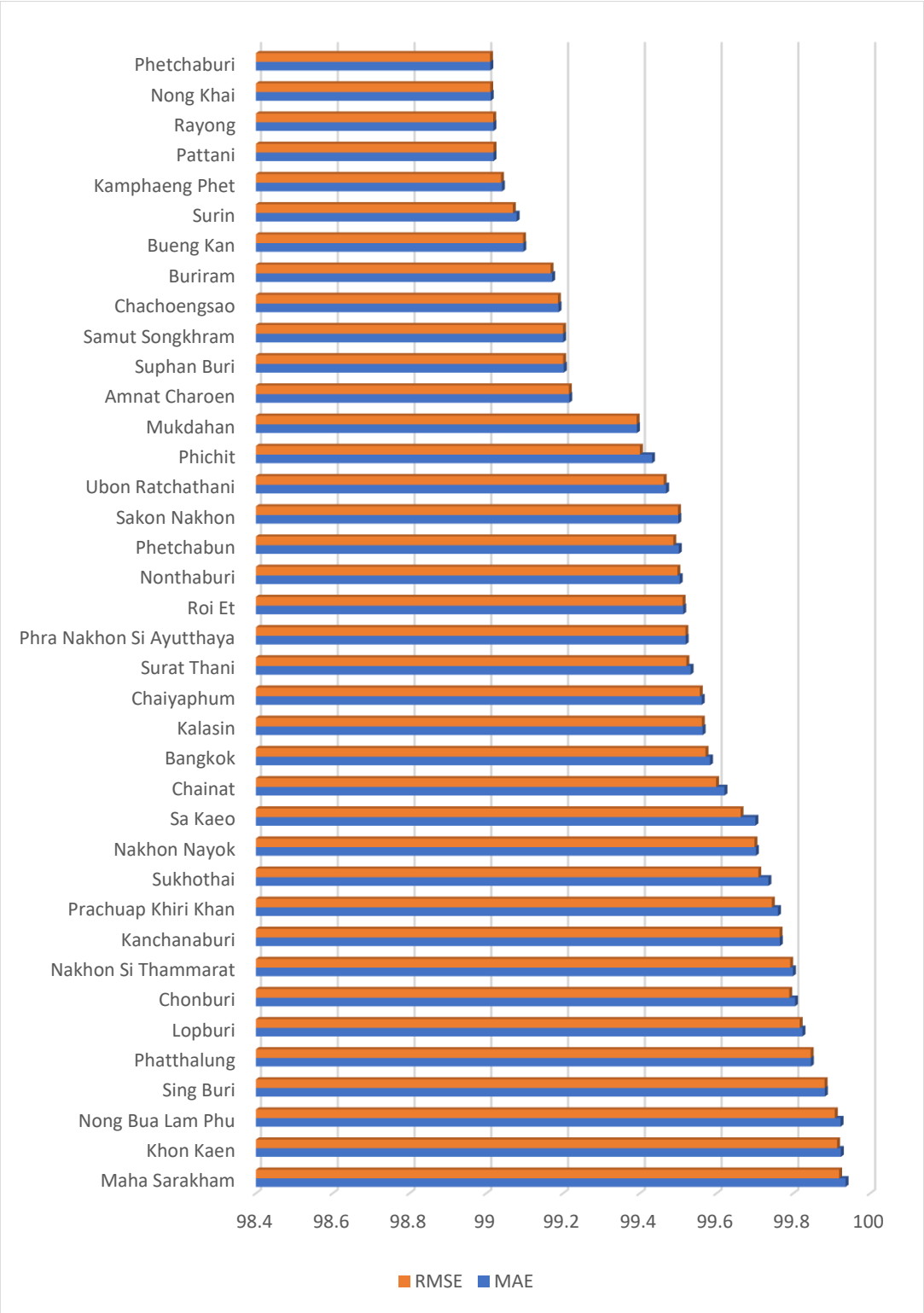
การทดสอบและประเมินประสิทธิภาพของโมเดลที่ผสมผสานระหว่าง LSTM และ MLPเป็นขั้นตอนสำคัญในการตรวจสอบความแม่นยำและความน่าเชื่อถือของโมเดลในการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 ในประเทศไทย ในส่วนนี้จะนำเสนอผลการทดลองที่ได้จากการใช้โมเดล LSTM + MLP โดยอ้างอิงจากภาพ 83 และตาราง 10 ซึ่งแสดงผลการทำนายจำนวนผู้ติดเชื้อสะสมและภาพ 84 และตาราง 11 ซึ่งแสดงผลการทำนายจำนวนผู้เสียชีวิตสะสม โดยตารางผลลัพธ์ที่ได้จากการวัดค่า MAE RMSE MSE

ของการติดเชื้อสะสมและการเสียชีวิตสะสมของโมเดล LSTM + MLP จังหวัดที่มีความแม่นยำมากที่สุดของการทำนายชุดข้อมูลการติดเชื่อสะสมได้แก่ จังหวัดมหาสารคามมีความถูกต้องร้อยละ 99.94 จังหวัดขอนแก่นมีความถูกต้องร้อยละ 99.92 จังหวัดหนองบัวลำภูมีความถูกต้องร้อยละ 99.92 จังหวัดสิงห์บุรีมีความถูกต้องร้อยละ 99.88 และจังหวัดพัทลุงมีความถูกต้องร้อยละ 99.85

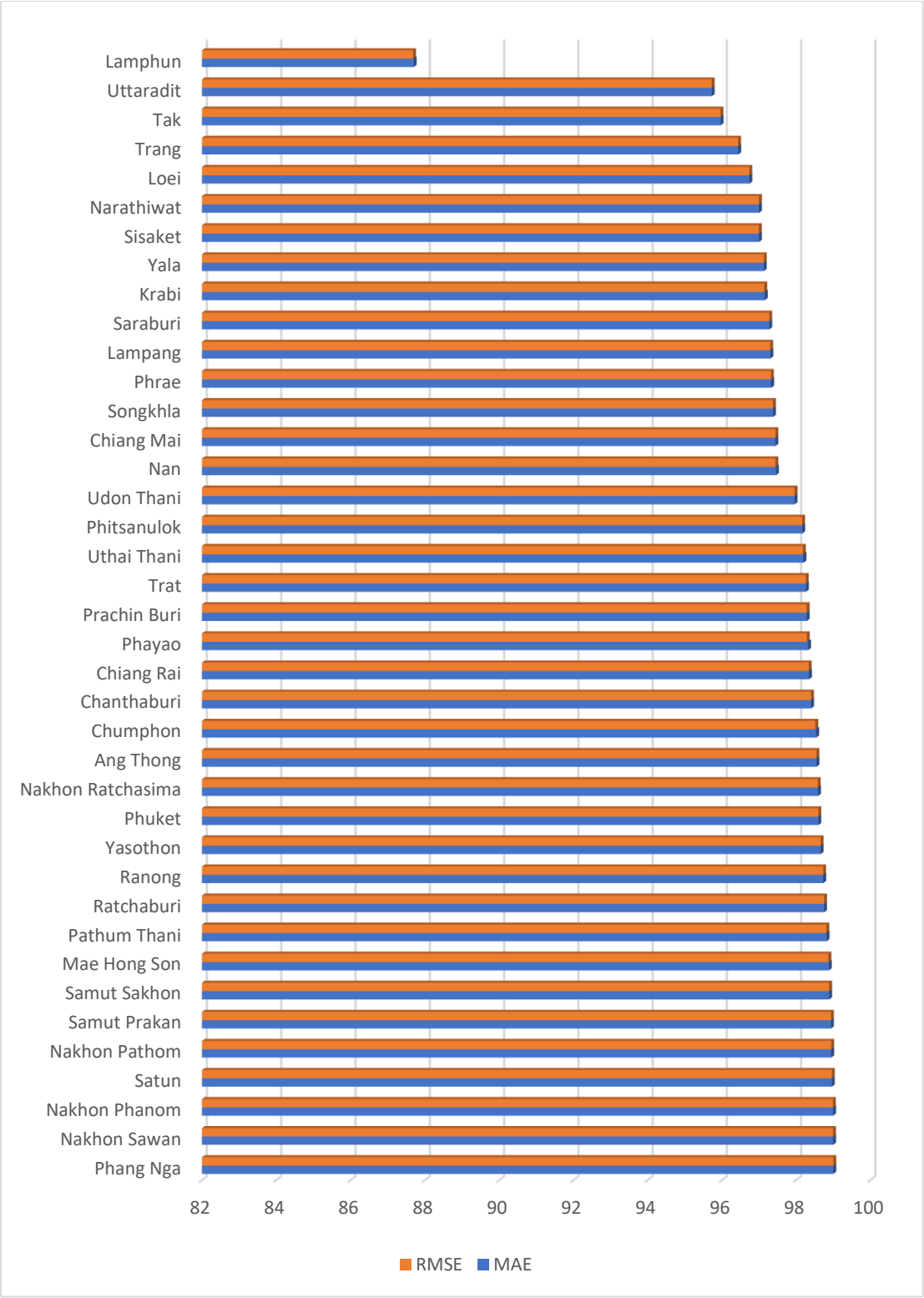
จังหวัดที่มีความแม่นยำน้อยที่สุดในการทำนายของโมเดล LSTM + MLP ในการทำนายชุดข้อมูลการติดเชื้อสะสมได้แก่ จังหวัดเลยมีความถูกต้องร้อยละ 96.74 จังหวัดตรังมีความถูกต้องร้อยละ 96.44 จังหวัดตากมีความถูกต้องร้อยละ 95.97 จังหวัดอุตรดิตถ์มีความถูกต้องร้อยละ 95.73 และจังหวัดลำพูนมีความถูกต้องร้อยละ 87.70

จังหวัดที่มีความแม่นยำสูงสุดในการทำนายชุดข้อมูลการเสียชีวิตสะสมของโมเดล LSTM + MLP ได้แก่ จังหวัดสมุทรสาครมีความถูกต้องร้อยละ 99.83 จังหวัดสระแก้วมีความถูกต้องร้อยละ 99.79 จังหวัดสมุทรสงครามมีความถูกต้องร้อยละ 99.78 จังหวัดประจวบคีรีขันธ์มีความถูกต้องร้อยละ 99.61 และจังหวัดบุรีรัมย์มีความถูกต้องร้อยละ 99.36

จังหวัดที่มีความแม่นยำน้อยที่สุดในการทำนายของโมเดล LSTM + MLP ในการทำนายชุดข้อมูลการเสียชีวิตสะสม ได้แก่ จังหวัดแพร่มีความถูกต้องร้อยละ 85.60 จังหวัดนครสวรรค์มีความถูกต้องร้อยละ 85.11 จังหวัดลำพูนมีความถูกต้องร้อยละ 82.11 จังหวัดพะเยามีความถูกต้องร้อยละ 81.91 และจังหวัดลำปางมีความถูกต้องร้อยละ 81.75



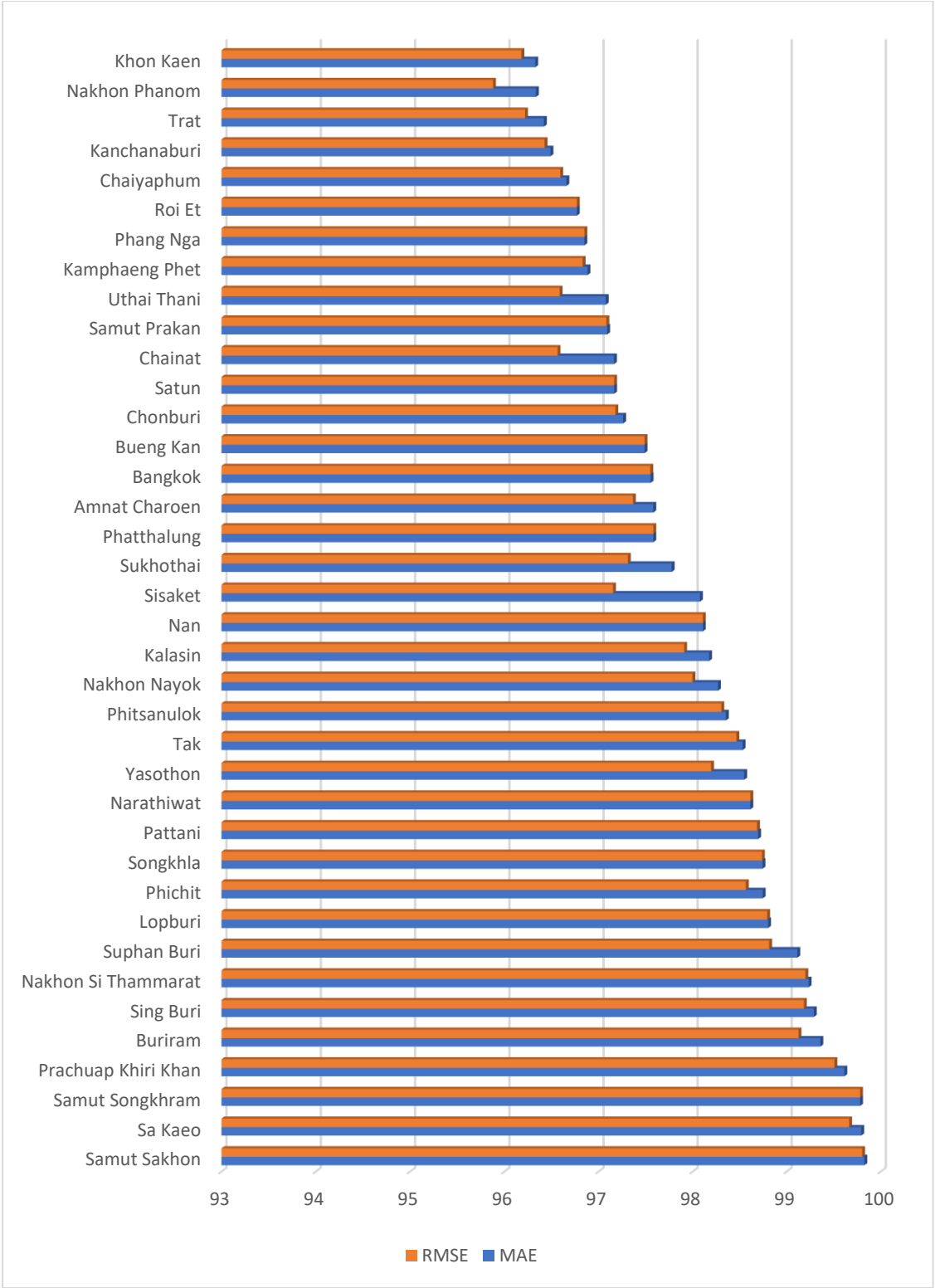
ภาพ 83 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM + MLP



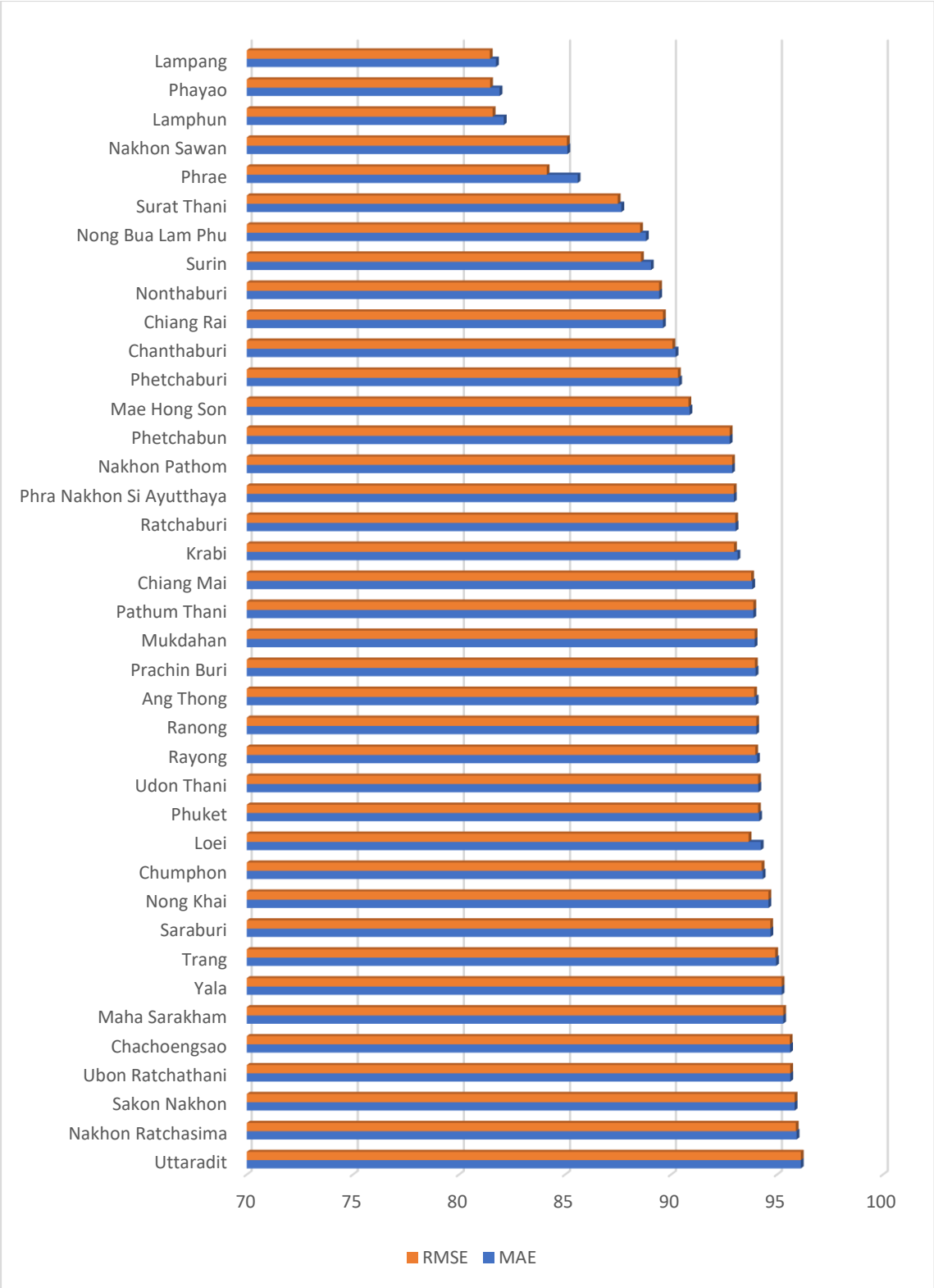
ภาพ 83 ผลการทำนายจำนวนผู้ติดเชื้อสะสมจากโมเดล LSTM + MLP (ต่อ)

ตาราง 10 ผลการทำนายของโมเดล LSTM+ MLP โดยใช้ชุดมูลการติดเชื้อสะสม

Province	ACC (MAE)	ACC (RMSE)	Province	ACC (MAE)	ACC (RMSE)
Maha Sarakham	99.94	99.92	Phang Nga	99.00	99.00
Khon Kaen	99.92	99.91	Nakhon Sawan	99.00	99.00
Nong Bua Lam Phu	99.92	99.91	Nakhon Phanom	99.00	98.99
Sing Buri	99.88	99.88	Satun	98.96	98.96
Phatthalung	99.85	99.85	Nakhon Pathom	98.95	98.95
Lopburi	99.82	99.82	Samut Prakan	98.94	98.94
Chonburi	99.80	99.79	Samut Sakhon	98.89	98.89
Nakhon Si Thammarat	99.80	99.79	Mae Hong Son	98.88	98.87
Kanchanaburi	99.77	99.76	Pathum Thani	98.82	98.82
Prachuap Khiri Khan	99.76	99.74	Ratchaburi	98.75	98.75
Sukhothai	99.74	99.71	Ranong	98.73	98.73
Nakhon Nayok	99.70	99.70	Yasothon	98.66	98.66
Sa Kaeo	99.70	99.66	Phuket	98.60	98.60
Chainat	99.62	99.60	Nakhon Ratchasima	98.59	98.58
Bangkok	99.58	99.57	Ang Thong	98.55	98.55
Kalasin	99.56	99.56	Chumphon	98.54	98.52
Chaiyaphum	99.56	99.56	Chanthaburi	98.41	98.40
Surat Thani	99.53	99.52	Chiang Rai	98.35	98.34
Phra Nakhon Si Ayutthaya	99.52	99.52	Phayao	98.33	98.29
Roi Et	99.51	99.51	Prachin Buri	98.29	98.29
Nonthaburi	99.50	99.50	Trat	98.27	98.26
Phetchabun	99.50	99.49	Uthai Thani	98.20	98.18
Sakon Nakhon	99.50	99.50	Phitsanulok	98.17	98.16
Ubon Ratchathani	99.47	99.46	Udon Thani	97.96	97.96
Phichit	99.43	99.40	Nan	97.46	97.44
Mukdahan	99.39	99.39	Chiang Mai	97.44	97.44
Amnat Charoen	99.22	99.22	Songkhla	97.38	97.38
Suphan Buri	99.20	99.20	Phrae	97.33	97.32
Samut Songkhram	99.20	99.20	Lampang	97.31	97.30
Chachoengsao	99.19	99.19	Saraburi	97.28	97.28
Buriram	99.17	99.17	Krabi	97.16	97.14
Bueng Kan	99.10	99.10	Yala	97.13	97.13
Surin	99.08	99.07	Sisaket	97.00	97.00
Kamphaeng Phet	99.04	99.04	Narathiwat	97.00	97.00
Pattani	99.02	99.02	Loei	96.74	96.74
Rayong	99.02	99.02	Trang	96.44	96.44
Nong Khai	99.01	99.01	Tak	95.97	95.96
Phetchaburi	99.01	99.01	Uttaradit	95.73	95.73
			Lamphun	87.70	87.68



ภาพ 84 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM + MLP



ภาพ 85 ผลการทำนายจำนวนผู้เสียชีวิตสะสมจากโมเดล LSTM + MLP (ต่อ)

ตาราง 11 ผลการทำนายของโมเดล LSTM + MLP โดยใช้ชุดมูลการเสียชีวิตสะสม

Province	ACC (MAE)	ACC (RMSE)	Province	ACC (MAE)	ACC (RMSE)
Samut Sakhon	99.83	99.80	Uttaradit	96.13	96.13
Sa Kaeo	99.79	99.66	Nakhon Ratchasima	95.93	95.89
Samut Songkhram	99.78	99.78	Sakon Nakhon	95.84	95.84
Prachuap Khiri Khan	99.61	99.51	Ubon Ratchathani	95.63	95.63
Buriram	99.36	99.13	Chachoengsao	95.62	95.61
Sing Buri	99.29	99.18	Maha Sarakham	95.30	95.30
Nakhon Si Thammarat	99.23	99.20	Yala	95.22	95.22
Suphan Buri	99.11	98.81	Trang	94.96	94.92
Lopburi	98.80	98.79	Saraburi	94.69	94.69
Phichit	98.75	98.57	Nong Khai	94.60	94.60
Songkhla	98.74	98.74	Chumphon	94.33	94.28
Pattani	98.70	98.68	Loei	94.24	93.66
Narathiwat	98.61	98.61	Phuket	94.16	94.11
Yasothon	98.55	98.20	Udon Thani	94.13	94.11
Tak	98.53	98.47	Rayong	94.06	93.98
Phitsanulok	98.35	98.31	Ranong	94.02	94.02
Nakhon Nayok	98.27	98.00	Ang Thong	94.00	93.92
Kalasin	98.18	97.91	Prachin Buri	93.99	93.96
Nan	98.11	98.11	Mukdahan	93.95	93.95
Sisaket	98.08	97.15	Pathum Thani	93.89	93.89
Sukhothai	97.78	97.31	Chiang Mai	93.84	93.78
Phatthalung	97.58	97.58	Krabi	93.14	92.98
Amnat Charoen	97.58	97.37	Ratchaburi	93.05	93.03
Bangkok	97.55	97.55	Phra Nakhon Si Ayutthaya	92.97	92.96
Bueng Kan	97.49	97.49	Nakhon Pathom	92.88	92.88
Chonburi	97.26	97.18	Phetchabun	92.77	92.77
Satun	97.17	97.17	Mae Hong Son	90.87	90.82
Chainat	97.17	96.57	Phetchaburi	90.38	90.33
Samut Prakan	97.09	97.08	Chanthaburi	90.22	90.07
Uthai Thani	97.08	96.59	Chiang Rai	89.62	89.62
Kamphaeng Phet	96.88	96.83	Nonthaburi	89.44	89.44
Phang Nga	96.85	96.85	Surin	89.05	88.58
Roi Et	96.77	96.77	Nong Bua Lam Phu	88.82	88.54
Chaiyaphum	96.66	96.59	Surat Thani	87.66	87.48
Kanchanaburi	96.49	96.43	Phrae	85.60	84.14
Trat	96.42	96.22	Nakhon Sawan	85.11	85.09
Nakhon Phanom	96.34	95.88	Lamphun	82.11	81.59
Khon Kaen	96.33	96.18	Phayao	81.91	81.47
			Lampang	81.75	81.46

ผลการทำนายของโมเดล LSTM +MLP ที่ใช้สำหรับการทำนายชุดข้อมูลผู้ติดเชื้อสะสมและการเสียชีวิตสะสมนั้นได้ทำการสรุปตามตาราง 12 ซึ่งเป็นการหาค่าเฉลี่ยร้อยละความถูกต้องของทุกจังหวัดที่โมเดลใช้ในการทำนาย

ตาราง 12 ผลการทดลองโมเดล LSTM + MLP

Models	Prediction	RMSE	MAE	ร้อยละความถูกต้อง	
				คำนวณจาก RMSE	คำนวณจาก MAE
LSTM + MLP	total cases	228.60	227.11	98.60	98.60
LSTM + MLP	total deaths	6.74	6.64	94.79	94.94

ตาราง 12 แสดงผลการทดลองของโมเดล LSTM + MLP ในการทำนายจำนวนผู้ติดเชื้อสะสมและผู้เสียชีวิตสะสมซึ่งสรุปจากตาราง 7 โดยใช้ตัวชี้วัด RMSE และ MAE เพื่อประเมินประสิทธิภาพของโมเดล ผลการทดลองนี้แสดงว่าโมเดล LSTM + MLP มีประสิทธิภาพสูงกว่าโมเดล LSTM แบบพื้นฐานอย่างชัดเจน ค่า RMSE และ MAE ของการทำนายจำนวนผู้ติดเชื้อสะสมอยู่ที่ 228.60 และ 227.11 ตามลำดับ โดยมีร้อยละความถูกต้องเท่ากับ 98.60 และ 98.60 ตามลำดับ ส่วนการทำนายจำนวนผู้เสียชีวิตสะสมมีค่า RMSE และ MAE อยู่ที่ 6.74 และ 6.64 ตามลำดับ โดยมีร้อยละความถูกต้องเท่ากับ 94.79 และ 94.94 ตามลำดับ

4.3 เปรียบเทียบประสิทธิภาพโมเดล

จากตาราง 9 ผลการทดลองของโมเดลที่ใช้ในการทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสมแสดงให้เห็นว่าโมเดล LSTM แบบพื้นฐานให้ค่าร้อยละความถูกต้องของโมเดลอ้างอิงจากค่า MAE ที่ 82.13 สำหรับการทำนายจำนวนผู้ติดเชื้อสะสม และ 62.56 สำหรับการทำนายจำนวนผู้เสียชีวิตสะสม ในการทดลองสร้างโมเดลเพื่อการทำนายนี้ โมเดลที่แสดงความแม่นยำสูงสุดคือ LSTM + MLP ซึ่งมีค่าร้อยละความถูกต้องอ้างอิงจากค่า MAE สำหรับการทำนายจำนวนผู้ติดเชื้อสะสมที่ 98.60 และการทำนายจำนวนผู้เสียชีวิตสะสมที่ 94.94 จากตาราง 13 การทดลองนี้ทำให้เห็นว่าโมเดล LSTM + MLP มีความแม่นยำมากกว่า LSTM แบบพื้นฐานอย่างชัดเจน และเมื่อพิจารณาประสิทธิภาพของโมเดลผ่าน Learning Curve พบว่า LSTM + MLP มีความเสถียรมากกว่า LSTM แบบพื้นฐานอย่างเด่นชัด เหตุผลที่ LSTM + MLP มีความแม่นยำสูงกว่านั้นเนื่องจากการเพิ่มชั้นของ

LSTM ที่ช่วยรองรับความซับซ้อนของข้อมูลที่เพิ่มขึ้น รวมถึงการใช้ MLP ในการจัดการข้อมูลที่หลากหลาย รวมถึงข้อมูลการฉีดวัคซีน ทำให้โมเดลสามารถจัดการกับข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น

ตาราง 13 ค่าเฉลี่ยของผลการทำนายและค่าร้อยละความถูกต้องของโมเดล

Models	Prediction	RMSE	MAE	ร้อยละความถูกต้อง	
				คำนวณจาก RMSE	คำนวณจาก MAE
LSTM แบบพื้นฐาน	total cases	4828.07	4827.80	82.13	82.13
LSTM แบบพื้นฐาน	total deaths	46.04	45.97	62.44	62.56
LSTM + MLP	total cases	228.60	227.11	98.60	98.60
LSTM + MLP	total deaths	6.74	6.64	94.79	94.94

เมื่อได้ผลลัพธ์จากการทดลองโมเดล LSTM ที่สามารถทำงานร่วมกันกับโมเดล MLP ได้แล้ว นั้นผลลัพธ์ของโมเดล นั้นจะนำไปแสดงผลที่เว็บแอปพลิเคชันที่ได้ออกแบบไว้จากโมดูลที่ 3 เพื่อให้รองรับการแสดงผลของการทำนายของโมเดลที่สามารถแสดงผลร่วมกับแผนที่ของประเทศไทย รวมไปถึงการแสดงผลสถิติและผลการทำนายการแพร่ระบาดของเชื้อโควิด-19 ในรูปแบบของกราฟ

4.4 ผลการพัฒนาเว็บแอปพลิเคชันเพื่อเป็นเครื่องมือการวิเคราะห์ทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19

ผลลัพธ์เว็บแอปพลิเคชันที่แสดงผลลัพธ์การทำนายโควิด-19 โดยทำหน้าที่คล้ายกับ Dashboard ที่สามารถ Interact กับผู้ใช้ได้ แอปพลิเคชันนี้พัฒนาด้วย React และประกอบไปด้วยสามส่วนหลักคือ แผนที่ กราฟแสดงสถิติ และกราฟแสดงการทำนาย ส่วนแผนที่แสดงพื้นที่แต่ละจังหวัดของประเทศไทย โดยพื้นที่ที่ได้ทำการเลือกจะแสดงเป็นสีเขียวเข้ม กราฟแสดงสถิติและกราฟแสดงการทำนายจะแสดงข้อมูลเพิ่มเติมเกี่ยวกับจำนวนผู้ติดเชื้อและผู้เสียชีวิตสะสม ทำให้ผู้ใช้สามารถติดตามและวิเคราะห์สถานการณ์ได้อย่างละเอียดและแม่นยำ

ภาพ 85 ตัวอย่างการเลือกแสดงผลจำนวนผู้ติดเชื้อสะสมของจังหวัดนครสวรรค์

ภาพ 86 ตัวอย่างการเลือกแสดงผลจำนวนผู้ติดเชื้อสะสมของจังหวัดสุราษฎร์ธานี

ภาพ 87 และ 88 เป็นการแสดงผลการทำนายจำนวนผู้ติดเชื้อสะสมในแต่ละจังหวัดของประเทศไทย ในภาพ 87 แสดงการทำนายจำนวนผู้ติดเชื้อสะสมของจังหวัดนครสวรรค์ ส่วนภาพ 88 แสดงการทำนายจำนวนผู้ติดเชื้อสะสมในจังหวัดสุราษฎร์ธานีกราฟด้านบนของทั้งสองภาพแสดงแนวโน้มการเพิ่มขึ้นของจำนวนผู้ติดเชื้อตามเวลา โดยมีจุดแดงที่แสดงเหตุการณ์สำคัญหรือการเปลี่ยนแปลงที่สำคัญของกราฟและกราฟด้านล่างแสดงการเปรียบเทียบระหว่างจำนวนผู้ติดเชื้อที่แท้จริงกับจำนวนที่คาดการณ์ในอนาคต การแสดงผลลัพธ์ในลักษณะนี้ช่วยให้ผู้ใช้สามารถติดตามและวิเคราะห์สถานการณ์ได้อย่างละเอียดและแม่นยำ

ภาพ 87 ตัวอย่างการเลือกแสดงผลจำนวนผู้เสียชีวิตสะสมของจังหวัดตราด

ภาพ 88 ตัวอย่างการเลือกแสดงผลจำนวนผู้เสียชีวิตสะสมของจังหวัดเชียงใหม่

ภาพ 89 และ 90 แสดงผลการทำนายจำนวนผู้เสียชีวิตสะสมในแต่ละจังหวัดของประเทศไทย ภาพ 89 แสดงการทำนายจำนวนผู้เสียชีวิตสะสมในจังหวัดตราดส่วนภาพ 90 แสดงการทำนายจำนวนผู้เสียชีวิตสะสมในจังหวัดเชียงใหม่ กราฟด้านบนของทั้งสองภาพแสดงแนวโน้มการเพิ่มขึ้นของจำนวนผู้เสียชีวิตตามเวลา โดยมีการใช้เส้นกราฟสีม่วงเพื่อแสดงแนวโน้มการเพิ่มขึ้นของจำนวนผู้เสียชีวิต กราฟด้านล่างแสดงการเปรียบเทียบระหว่างจำนวนผู้เสียชีวิตที่แท้จริงกับจำนวนที่คาดการณ์ในอนาคต

4.5 อธิปรายผลการทดลอง

4.5.1 อภิปรายโมเดล

ในส่วนของการอภิปรายเกี่ยวกับโมเดล LSTM ที่นำมาใช้ในการพยากรณ์สถานการณ์การแพร่ระบาดของโรคโควิด-19 นั้น โมเดล LSTM มีจุดแข็งในการจัดการกับข้อมูลที่เป็นอนุกรมเวลา โดยสามารถเก็บข้อมูลจากอดีตและนำมาใช้ในการพยากรณ์ข้อมูลในอนาคตได้อย่างมีประสิทธิภาพ

อย่างไรก็ตาม LSTM ยังมีข้อจำกัดในด้านการจัดการกับข้อมูลที่ไม่ใช่ข้อมูลแบบอนุกรมเวลา ซึ่งทำให้การพยากรณ์โดยใช้โมเดล LSTM เพียงอย่างเดียวยังขาดความหลากหลายของข้อมูลที่อาจมีผลต่อความแม่นยำของการพยากรณ์เนื่องจากข้อมูลที่มีอยู่ในสถานการณ์การแพร่ระบาดของโรคโควิด-19 นั้นประกอบไปด้วยข้อมูลที่หลากหลาย ทั้งข้อมูลที่เป็นอนุกรมเวลา เช่น จำนวนผู้ติดเชื้อสะสม และข้อมูลแบบคงที่ เช่น ข้อมูลความหนาแน่นประชากร การใช้โมเดล LSTM เพียงอย่างเดียวจึงไม่สามารถครอบคลุมการวิเคราะห์ข้อมูลทั้งหมดได้อย่างครบถ้วน ด้วยเหตุนี้ การนำโมเดล MLP เข้ามาผสมผสานกับ LSTM จึงเป็นการเพิ่มความสามารถในการจัดการกับข้อมูลที่หลากหลายยิ่งขึ้น

MLP ได้นำมาใช้เพื่อจัดการกับข้อมูลแบบคงที่ ซึ่งเป็นข้อมูลที่ไม่ได้ขึ้นอยู่กับลำดับเวลา การผสมผสานระหว่าง LSTM และ MLP ทำให้โมเดลที่พัฒนาขึ้นมีความยืดหยุ่นในการจัดการกับข้อมูลทั้งสองประเภท และสามารถนำข้อมูลทั้งแบบอนุกรมเวลาและแบบคงที่มาใช้ในการพยากรณ์ได้อย่างมีประสิทธิภาพมากขึ้น การที่ MLP สามารถจัดการกับข้อมูลที่ไม่เป็นลำดับเวลาได้ดีนั้น ช่วยเสริมสร้างความสามารถของโมเดลในการพยากรณ์สถานการณ์ที่ซับซ้อนมากขึ้น โดยไม่จำกัดเพียงแค่การวิเคราะห์ข้อมูลที่เป็นอนุกรมเวลาเพียงอย่างเดียว การผสมผสานนี้จึงทำให้การพยากรณ์มีความถูกต้องและแม่นยำมากยิ่งขึ้น

การศึกษานี้มีจุดมุ่งหมายเพื่อพัฒนาโมเดลการเรียนรู้เชิงลึกเพื่อทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 โดยใช้โมเดล LSTM และ MLP ซึ่งได้รับการออกแบบมาเพื่อเพิ่มความแม่นยำในการทำนายการใช้โมเดล LSTM ช่วยให้สามารถจัดการกับข้อมูลที่มีลำดับเวลาได้อย่างมีประสิทธิภาพ ขณะที่โมเดล MLP ใช้เพื่อประมวลผลข้อมูลแบบคงที่ทั้งสองโมเดลนำมารวมกันเพื่อเสริมความสามารถในการทำนายโดยโมเดล MLP นั้นเข้ามามีบทบาทในการช่วยให้โมเดล LSTM สามารถเขียนรู้ข้อมูลที่มีความหลากหลายมากขึ้นนอกจากข้อมูลที่เป็นแบบอนุกรมเวลาอย่างเดียว

การอภิปรายในย่อหน้าถัดไปเป็นการอภิปรายการนำผลการรทำนายของโมเดล LSTM แบบพื้นฐานและโมเดล LSTM + MLP ของชุดข้อมูลการติดเชื้อสะสมมาทำการอภิปรายโดยเลือกจากห้าอันดับจังหวัดที่มีความแม่นยำน้อยที่สุดของแต่ละโมเดลมาทำการอภิปรายจากตารางร้อยละความถูกต้องของแต่ละโมเดลดังตาราง 14

ตาราง 14 ผลห้าจังหวัดที่มีความแม่นยำน้อยที่สุดของโมเดล LSTM แบบพื้นฐาน

จังหวัด	ร้อยละความถูกต้องคำนวณจาก MAE	ร้อยละความถูกต้องคำนวณจาก RMSE
เชียงราย	65.40	65.39
กรุงเทพฯ	60.04	60.04
สุพรรณบุรี	59.21	59.21
ศรีสะเกษ	55.13	55.13
หนองคาย	49.42	49.42

จากตาราง 14 แสดงค่าร้อยละความถูกต้องของโมเดล LSTM แบบพื้นฐาน โดยพิจารณาจากค่า MAE ของชุดข้อมูลการติเชื้อสะสม พบว่าจังหวัดที่โมเดลมีความแม่นยำต่ำที่สุดคือหนองคาย โดยมีค่าร้อยละความถูกต้องของโมเดลอยู่ที่ 49.42 ซึ่งบ่งบอกว่าโมเดลไม่สามารถทำนายข้อมูลได้อย่างถูกต้อง ลำดับถัดไปคือจังหวัดศรีสะเกษที่มีค่าร้อยละความถูกต้องอยู่ที่ 55.13 และจังหวัดสุพรรณบุรีที่มีค่าร้อยละความถูกต้องอยู่ที่ 59.21 ซึ่งแม้ว่าจะมีความแม่นยำสูงขึ้นเล็กน้อย ในขณะที่กรุงเทพมหานคร และเชียงรายมีค่าร้อยละความถูกต้องอยู่ที่ 60.04 และ 65.39 ตามลำดับ ซึ่งแสดงว่าโมเดลสามารถทำนายผลได้ดีกว่าในสองจังหวัดนี้ แต่ก็ยังมีความผิดพลาดอยู่ ปัญหาที่ทำให้เกิดความผิดพลาดเหล่านี้อาจเป็นผลมาจากการ Overfitting ของโมเดล ซึ่งทำให้โมเดลจดจำรายละเอียดเฉพาะของข้อมูลในการฝึกได้ดีเกินไป จนไม่สามารถทั่วไปข้อมูลได้เมื่อเจอกับข้อมูลใหม่ หรืออาจเกิดจากการที่โมเดลไม่สามารถจับความซับซ้อนของข้อมูลในแต่ละจังหวัดได้อย่างถูกต้อง

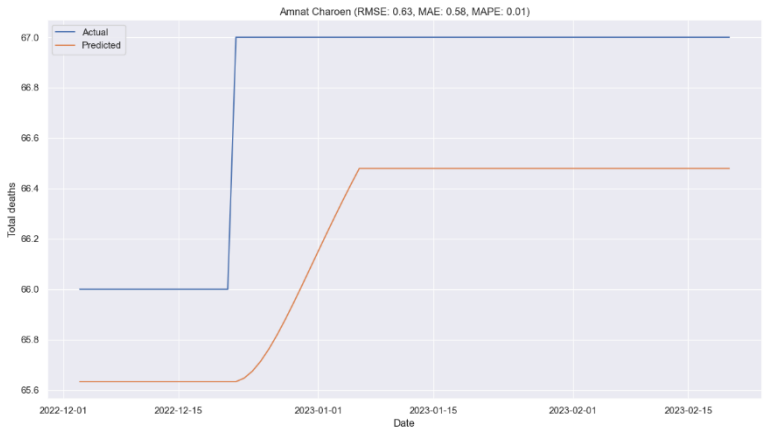
ตาราง 15 ผลห้าจังหวัดที่มีความแม่นยำน้อยที่สุดของโมเดล LSTM + MLP

จังหวัด	ร้อยละความถูกต้องคำนวณจาก MAE	ร้อยละความถูกต้องคำนวณจาก RMSE
เลย	96.74	96.74
ตรัง	96.44	96.44
ตาก	95.97	95.96
อุตรดิตถ์	95.73	95.73
ลำพูน	87.70	87.68

จากตาราง 15 แสดงค่าร้อยละความถูกต้องของโมเดล LSTM แบบพื้นฐานในการทำนายข้อมูลการติดเชื้อสะสม พบว่าจังหวัดลำพูนมีค่าความถูกต้องต่ำที่สุดในกลุ่มนี้ โดยมีค่าร้อยละความถูกต้องอยู่ที่ 87.70 ซึ่งบ่งชี้ว่าโมเดลมีความแม่นยำน้อยที่สุดในการทำนายข้อมูลเทียบกับจังหวัดอื่น ๆ ที่มีค่าความถูกต้องสูงกว่า เช่น อุตรดิตถ์และตากที่มีค่าร้อยละความถูกต้องอยู่ที่ 95.72 และ 95.96 ตามลำดับ การที่โมเดลมีความแม่นยำต่ำสุดในจังหวัดลำพูนอาจเป็นเพราะโมเดลไม่สามารถจับความซับซ้อนของข้อมูลในพื้นที่นี้ได้ดีเท่ากับจังหวัดอื่น ๆ ข้อมูลในลำพูนมีความซับซ้อนเฉพาะหรือมีลักษณะที่แตกต่างจากข้อมูลในจังหวัดอื่น ๆ ทำให้โมเดลไม่สามารถเรียนรู้และทำนายผลได้อย่างแม่นยำ ในขณะเดียวกันจังหวัดที่โมเดลมีความแม่นยำมากที่สุดคือจังหวัดเลย โดยมีค่าร้อยละความถูกต้องสูงถึง 96.74 ซึ่งหมายความว่าโมเดลสามารถจับรูปแบบและความซับซ้อนของข้อมูลในจังหวัดเลยได้ดีกว่าเมื่อเทียบกับจังหวัดอื่น ๆ ที่แสดงในตาราง 15 การที่โมเดลสามารถทำนายได้แม่นยำในจังหวัดเหล่านี้อาจเป็นเพราะข้อมูลในพื้นที่มีความสม่ำเสมอหรือมีรูปแบบที่โมเดลสามารถเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้น ข้อมูลที่มีความซับซ้อนน้อยหรือความผันผวนที่ต่ำกว่าก็อาจเป็นปัจจัยสำคัญที่ช่วยให้โมเดลสามารถทำนายได้อย่างแม่นยำในจังหวัดเหล่านี้.

อีกหนึ่งปัญหาที่สำคัญในการพัฒนาโมเดลคือปัญหา Lag Of Prediction หรือความล่าช้าในการพยากรณ์ของโมเดล เกิดจากการที่โมเดลไม่สามารถคาดการณ์ผลลัพธ์ในอนาคตได้อย่างทันที เนื่องจาก LSTM ใช้ข้อมูลจากอดีตเพื่อทำนายอนาคต ซึ่งอาจทำให้เกิดความล่าช้าเมื่อข้อมูลที่ส่งเข้าสู่โมเดลมีความหลากหลายหรือซับซ้อน และการปรับตัวของโมเดลให้สามารถจับแนวโน้มและลักษณะของข้อมูลที่เปลี่ยนแปลงอยู่ตลอดเวลา

ในการทดลองนี้ใช้โมเดลแบบ LSTM ร่วมกับ MLP เพื่อทำนายจำนวนผู้เสียชีวิตสะสม ผู้วิจัยได้เลือกตัวอย่างหนึ่งในจังหวัดที่เกิดความล่าช้า (Lag) ในการทำนายของโมเดล จากกราฟแสดงผลการทำนายของจังหวัดอำนาจเจริญที่มีค่าความแม่นยำที่ร้อยละของ RMSE และ MAE อยู่ที่ 99.22 จากการวิเคราะห์กราฟการทำนายของจังหวัดอำนาจเจริญพบว่ามีปัญหาความล่าช้าในการทำนายเมื่อเปรียบเทียบกับข้อมูลจริงดังภาพ 91 โดยเฉพาะในช่วงที่มีการเปลี่ยนแปลงจำนวนผู้เสียชีวิตอย่างรวดเร็ว โมเดลไม่สามารถตอบสนองต่อการเปลี่ยนแปลงนี้ได้ทันที ซึ่งแสดงให้เห็นว่าโมเดลอาจมีความยากลำบากในการจับลักษณะการเปลี่ยนแปลงที่รวดเร็วในข้อมูลจริง



ภาพ 89 ความล่าช้าที่เกิดจากการทำนายของโมเดล LSTM + MLP ของจังหวัดอำนาจเจริญ

ปัญหานี้อาจเกิดจากหลายสาเหตุ หนึ่งในสาเหตุหลักคือ LSTM ซึ่งแม้ว่าจะสามารถจัดการกับข้อมูลลำดับเวลาและระลึกถึงข้อมูลระยะยาวได้ดี แต่บางครั้งก็อาจตอบสนองช้าในกรณีที่ข้อมูลจริงมีการเปลี่ยนแปลงอย่างรวดเร็ว นอกจากนี้การรวม MLP อาจไม่ได้เพิ่มประสิทธิภาพในการทำนายช่วงเวลาที่เกิดการเปลี่ยนแปลงอย่างฉับพลัน เนื่องจาก MLP นั้นไม่ได้ออกแบบมาเพื่อจัดการกับข้อมูลลำดับเวลาโดยตรง ดังนั้นการรวมโมเดลทั้งสองอาจทำให้เกิดการตอบสนองที่ช้า อีกหนึ่งสาเหตุที่อาจทำให้เกิดปัญหาความล่าช้าในการทำนาย คือการที่โมเดลอาจไม่ได้รับข้อมูลหรือฟีเจอร์ที่มีความสำคัญเพียงพอในการช่วยให้โมเดลสามารถจับการเปลี่ยนแปลงที่เกิดขึ้นในข้อมูลได้ การเลือกฟีเจอร์หรือการประมวลผลข้อมูลล่วงหน้า ข้อมูลที่ไม่เหมาะสมอาจทำให้โมเดลไม่ได้รับข้อมูลที่เป็นประโยชน์ต่อการทำนายที่แม่นยำ ซึ่งอาจส่งผลให้โมเดลมีการตอบสนองช้าหรือไม่แม่นยำเมื่อเกิดการเปลี่ยนแปลงในข้อมูลจริง

ดังนั้นในการปรับปรุงโมเดลครั้งต่อไป เพื่อแก้ไขปัญหาเหล่านี้ หนึ่งในเทคนิคที่สามารถนำมาใช้คือการทำ shifting หรือการเลื่อนข้อมูลการทำนายไปข้างหน้าเพื่อให้โมเดลสามารถทำนายการเปลี่ยนแปลงได้เร็วขึ้น วิธีนี้จะช่วยลดความล่าช้าในการทำนาย เนื่องจากโมเดลจะฝึกให้ทำนายล่วงหน้าจากข้อมูลที่เกิดขึ้นจริงในช่วงก่อนหน้านั้น นอกจากนี้ยังสามารถพิจารณาการใช้เทคนิคอื่นๆ เช่น Temporal Ensembling ที่รวมเอาผลลัพธ์จากหลายๆ ช่วงเวลามาทำการทำนายร่วมกัน

4.5.2 อภิปรายเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันที่พัฒนาขึ้นโดยสามารถแสดงผลข้อมูลในรูปแบบแผนที่ที่มีความละเอียดในระดับจังหวัด เป็นข้อดีที่สำคัญในด้านการวิเคราะห์และการติดตามสถานการณ์การแพร่ระบาดของโรคโควิด-19 ในประเทศไทย การแสดงผลผ่านแผนที่ในลักษณะนี้ช่วยให้ผู้ใช้สามารถสังเกตเห็นการแพร่กระจายของโรคในแต่ละพื้นที่ได้อย่างชัดเจนและเข้าใจง่าย นอกจากนี้ การที่แผนที่สามารถแยกแยะรายละเอียดในระดับจังหวัดยังสนับสนุนการวิเคราะห์และการตัดสินใจที่มีประสิทธิภาพในการรับมือกับสถานการณ์ที่เกิดขึ้น เนื่องจากผู้ใช้งานสามารถมองเห็นภาพรวมของการระบาดได้อย่างครอบคลุมและเชื่อมโยงกับข้อมูลทางภูมิศาสตร์ ซึ่งเป็นเครื่องมือที่สำคัญในการวิเคราะห์เชิงพื้นที่และการคาดการณ์ความเสี่ยงของการระบาดในอนาคต

นอกจากนี้เว็บแอปพลิเคชันที่ได้ออกแบบยังมีความสามารถในการแสดงกราฟข้อมูลจริงเกี่ยวกับการติดเชื้อสะสมและการเสียชีวิตสะสมได้อย่างชัดเจน ซึ่งช่วยให้ผู้ใช้งานสามารถวิเคราะห์แนวโน้มและรูปแบบของการแพร่ระบาดในแต่ละช่วงเวลาได้ การที่แอปพลิเคชันมีแถบเลื่อนเพื่อให้ผู้ใช้สามารถเลือกช่วงเวลาที่ต้องการวิเคราะห์ได้ ช่วยเพิ่มความยืดหยุ่นในการตรวจสอบและเปรียบเทียบข้อมูลในช่วงเวลาต่าง ๆ ได้อย่างละเอียด การแสดงผลข้อมูลในรูปแบบกราฟเช่นนี้นอกจากจะช่วยให้การวิเคราะห์ข้อมูลเป็นไปอย่างง่ายดายแล้ว ยังเพิ่มความชัดเจนในการเปรียบเทียบข้อมูลระหว่างช่วงเวลาหรือระหว่างพื้นที่ต่าง ๆ ซึ่งมีความสำคัญอย่างยิ่งต่อการควบคุมและป้องกันการแพร่ระบาดของโรค

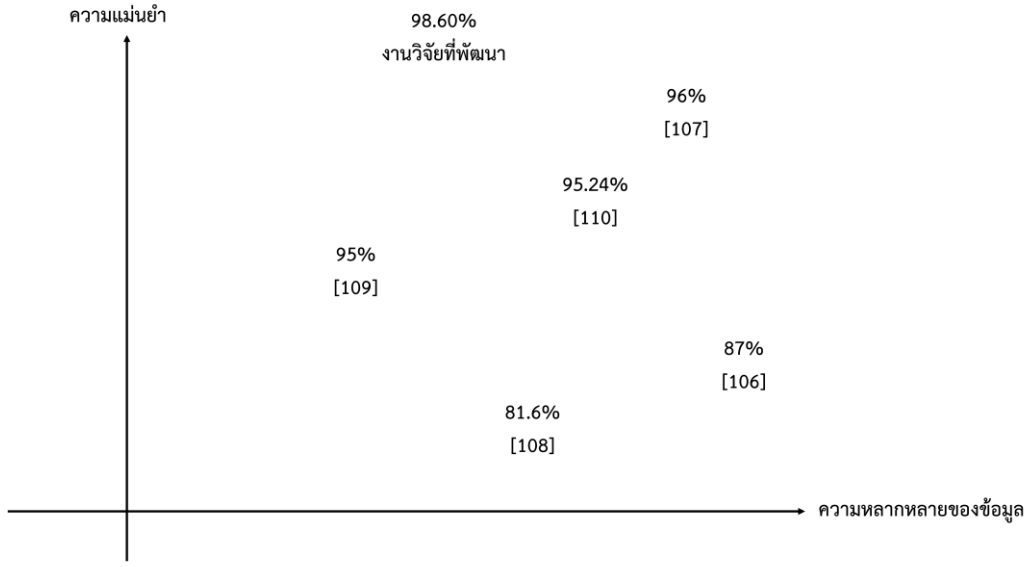
การพัฒนาเว็บแอปพลิเคชันเพื่อแสดงผลการทำนายด้วยโมเดล LSTM + MLP ยังมีหลายจุดที่ควรได้รับการปรับปรุงเพื่อให้เว็บแอปพลิเคชันนี้สามารถตอบสนองต่อความต้องการของผู้ใช้งานได้ดียิ่งขึ้น เริ่มจากการออกแบบ User Interface (UI) ที่สามารถทำให้เข้าใจง่ายขึ้น ปัจจุบัน UI อาจมีความซับซ้อน และการจัดวางองค์ประกอบบางอย่างอาจทำให้ผู้ใช้รู้สึกสับสนและเข้าใจยาก การจัดเรียงข้อมูลให้เป็นระบบระเบียบ และใช้สีหรือสัญลักษณ์ที่ชัดเจน สามารถช่วยให้ผู้ใช้สามารถตีความข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น นอกจากนี้ ความสามารถในการโต้ตอบกับข้อมูลในเว็บแอปพลิเคชันยังสามารถพัฒนาเพิ่มเติมได้ การเพิ่มฟีเจอร์ที่ให้ผู้ใช้สามารถเลือกช่วงเวลาหรือพื้นที่ที่ต้องการดูข้อมูลได้ จะช่วยให้การใช้งานมีความยืดหยุ่นและตรงกับความต้องการของผู้ใช้มากยิ่งขึ้น ฟีเจอร์เสริม เช่น ระบบแจ้งเตือนเมื่อมีการเปลี่ยนแปลงที่สำคัญในข้อมูล หรือการให้ผู้ใช้สามารถดาวน์โหลดข้อมูลในรูปแบบต่าง ๆ เช่น CSV หรือ Excel ก็เป็นสิ่งที่ควรพิจารณาเพิ่มเติม เพราะจะทำให้ผู้ใช้สามารถนำข้อมูลไปใช้งานต่อได้สะดวกยิ่งขึ้น

นอกจากการปรับปรุงด้านการออกแบบ UI และเพิ่มฟีเจอร์ให้แอปพลิเคชันมีความยืดหยุ่นมากขึ้นแล้ว อีกหนึ่งส่วนที่ควรพิจารณาเพิ่มเติมคือการรวมข้อมูลสถิติในด้านอื่น ๆ ที่เกี่ยวข้อง เช่น ข้อมูลการฉีดวัคซีน จำนวนผู้ติดเชื้อในแต่ละวัน หรือข้อมูลประชากร ซึ่งข้อมูลเหล่านี้สามารถช่วยให้

ผู้ใช้เห็นภาพรวมที่ชัดเจนขึ้นและนำไปสู่การวิเคราะห์ที่มีประสิทธิภาพมากขึ้น การรวมข้อมูลหลายมิติจะช่วยให้ผู้ใช้สามารถดูความสัมพันธ์ระหว่างข้อมูลการทำนายกับปัจจัยอื่น ๆ ได้ดีขึ้น เช่น การดูความสัมพันธ์ระหว่างจำนวนผู้เสียชีวิตสะสมกับอัตราการฉีดวัคซีน หรือการวิเคราะห์ผลกระทบของการระบาดในพื้นที่ต่าง ๆ

4.5.3 การเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

ในการอภิปรายเกี่ยวกับโมเดล LSTM ที่พัฒนาขึ้นในงานวิจัยนี้ ได้มีการเปรียบเทียบกับงานวิจัยอื่น ๆ อีก 5 งาน คือ [106-110] โดยใช้เกณฑ์การประเมินสองมิติ คือ ความแม่นยำของโมเดล และความหลากหลายของข้อมูลที่ใช้ในการวิเคราะห์ ผลการเปรียบเทียบดังกล่าวแสดงให้เห็นในภาพ 92



ภาพ 90 การเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

จากผลการเปรียบเทียบภาพ 92 เห็นได้ว่างานวิจัยที่พัฒนามีความแม่นยำสูงสุดที่ร้อยละ 98.60 ซึ่งบ่งบอกถึงความสามารถของโมเดล LSTM ในการพยากรณ์ข้อมูลอนุกรมเวลาได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม เมื่อพิจารณาในมิติของความหลากหลายของข้อมูลที่ใช้ในการวิเคราะห์ งานวิจัยที่พัฒนายังคงด้อยกว่างานวิจัยอื่น ๆ อีก 4 งานที่นำมาเปรียบเทียบ โดยเฉพาะ 4 งานวิจัยที่

ใช้ข้อมูลที่ค่อนข้างหลากหลายซึ่งครอบคลุมเกี่ยวกับปัจจัยการแพร่ระบาดของเชื้อโควิด-19 ที่มากกว่า ทำให้การวิเคราะห์ปัจจัยต่างๆได้มากกว่า

แม้ว่างานวิจัยที่พัฒนาจะมีความโดดเด่นในด้านความแม่นยำ แต่การที่ความหลากหลายของข้อมูลยังจำกัดอยู่ในกรอบของข้อมูลอนุกรมเวลา ทำให้ไม่สามารถครอบคลุมมิติอื่น ๆ ที่มีความสำคัญต่อการพยากรณ์ได้อย่างเต็มที่ ในงานวิจัยนี้ การเพิ่มโมเดล MLP จึงนำมาใช้เพื่อเสริมศักยภาพในการจัดการกับข้อมูลแบบคงที่ที่หลากหลายยิ่งขึ้น อย่างไรก็ตาม แม้จะมีการรวม MLP เข้ามา แต่ความหลากหลายของข้อมูลที่ใช้ในการพยากรณ์ยังคงน้อยกว่าที่ปรากฏในงานวิจัยอื่น ๆ เพื่อให้การพัฒนาต่อยอดจากงานวิจัยนี้สามารถเพิ่มประสิทธิภาพ จำเป็นต้องพิจารณาการรวมข้อมูลที่มีความหลากหลายมากขึ้นในกระบวนการวิเคราะห์และพยากรณ์ โดยเฉพาะอย่างยิ่งการผสมผสานระหว่างข้อมูลอนุกรมเวลาและข้อมูลแบบคงที่ เพื่อให้ได้โมเดลที่สามารถตอบสนองต่อสถานการณ์ที่ซับซ้อนได้อย่างครอบคลุมและแม่นยำยิ่งขึ้น ซึ่งจะช่วยเพิ่มความน่าเชื่อถือในการประยุกต์ใช้งานโมเดลในบริบทของการวิเคราะห์เชิงปฏิบัติที่มีความซับซ้อนในอนาคต

จากตาราง 16 เมื่อเปรียบเทียบโมเดลที่นำเสนอกับโมเดลงานวิจัยที่เกี่ยวข้องกันแล้วพบว่าโมเดลที่ได้ทำการนำเสนอนั้นมีความแม่นยำกว่างานวิจัยที่เกี่ยวข้องทั้งหมดที่ได้นำมาเปรียบเทียบ โดยในแต่ละงานวิจัยนั้นใช้โมเดลของการเรียนรู้เชิงลึกที่มีความแตกต่างกันออกไปและที่สำคัญในแต่ละโมเดลของแต่ละงานวิจัยนั้นใช้ชุดข้อมูลที่แตกต่างกันด้วยเช่นกันโดยผู้วิจัยต่างมุ่งเน้นไปในการใช้โมเดลของการเรียนรู้เพื่อการทำนายการแพร่กระจายของเชื้อโควิด 19 ภายในประเทศของผู้วิจัยซึ่งเป็นผลทำให้แต่งานวิจัยที่นำมาเปรียบเทียบดังตาราง 16 ใช้ชุดข้อมูที่มีความแตกต่างกันและสาเหตุที่ทำให้แต่ละงานวิจัยที่ใช้ในการทำนายการแพร่ระบาดของเชื้อโควิด19 มีความแม่นยำที่ต่างกันคือการเลือกใช้โมเดลหรืออัลกอริทึมที่แตกต่างกันด้วยเช่นกันเนื่องจากในแต่ละงานวิจัยใช้ชุดข้อมูลในการฝึกอบรมที่แตกต่างกันจึงจำเป็นต้องเลือกโมเดลที่เหมาะสมกับชุดข้อมูลที่มีอยู่ของแต่ละงานวิจัย

ตาราง 16 การเปรียบเทียบความถูกต้องระหว่างงานวิจัย

โมเดล	โมเดลของงานวิจัยที่เกี่ยวข้อง	ความแม่นยำ	แผนที่
[106]	•SUPER LEARNER ENSEMBLE	87%	มี
[107]	•LSTM + MARKOV	96%	มี
[108]	•RBF •FCM •NARX	81.6%	มี
[109]	•GPR •SVM •DECISION TREE	95%	มี

[110]	•BAYESIAN •ADABOOST •SVM •KNN •RANDOM FOREST •C4.5 •ANN	95.24%	มี
โมเดลที่นำเสนอ	LSTM+MLP	98.60%	มี

บทที่ 5
สรุปผลการทดลอง

สถานการณ์การแพร่ระบาดของโควิด-19 ในประเทศไทยยังคงมีผู้ติดเชื้อและผู้เสียชีวิตเพิ่มขึ้นอย่างต่อเนื่อง งานวิจัยนี้จึงได้พัฒนาเครื่องมือเพื่อช่วยวิเคราะห์สถานการณ์โควิด-19 ในประเทศไทย โดยได้พัฒนาโมเดลเพื่อทำนายจำนวนผู้ติดเชื้อและผู้เสียชีวิตจากโควิด-19 และแสดงผลข้อมูลการทำนายในรูปแบบเว็บแอปพลิเคชั่น

5.1 ผลการพัฒนาโมเดล

การสร้างเครื่องมือในการทำนายการแพร่ระบาดของโควิด-19 ใช้เทคนิคการเรียนรู้ของเครื่องเชิงลึกร่วมกับระบบข้อมูลภูมิสารสนเทศ (GIS) เพื่อทำนายจำนวนผู้ติดเชื้อสะสมและจำนวนผู้เสียชีวิตสะสมของทั้ง 77 จังหวัดในประเทศไทย โมเดลที่นำเสนอคือ LSTM + MLP ซึ่งเป็นการทำงานร่วมกันของโมเดล LSTM และโมเดล MLP โดยที่โมเดล LSTM ใช้กับข้อมูลอนุกรมเวลาและโมเดล MLP ใช้กับข้อมูลแบบคงที่ เปรียบเทียบกับโมเดล LSTM แบบพื้นฐาน ผลลัพธ์แสดงให้เห็นว่าโมเดลที่ดีที่สุดคือการผสานระหว่าง LSTM และ MLP ซึ่งมีความแม่นยำในการทำนายจำนวนผู้ติดเชื้อสะสมถึงร้อยละ 98.60 โดยอ้างอิงจากค่า MAE และการทำนายจำนวนผู้เสียชีวิตสะสมมีความแม่นยำที่ร้อยละ 94.94 โดยอ้างอิงจากค่า MAE เช่นกัน

โมเดล LSTM + MLP ยังแสดงประสิทธิภาพดีในการลดค่า RMSE โดยค่า RMSE ของโมเดล LSTM + MLP ในการทำนายจำนวนผู้ติดเชื้อสะสมเท่ากับ 228.60 และ MAE เท่ากับ 227.11 ขณะที่โมเดล LSTM แบบพื้นฐานมีค่า RMSE เท่ากับ 4828.07 และ MAE เท่ากับ 4827.80 สำหรับการทำนายจำนวนผู้เสียชีวิตสะสม โมเดล LSTM + MLP มีค่า RMSE เท่ากับ 6.74 และ MAE เท่ากับ 6.64 ในขณะที่โมเดล LSTM แบบพื้นฐานมีค่า RMSE เท่ากับ 46.04 และ MAE เท่ากับ 45.97 ผลการทดลองเหล่านี้ยืนยันว่าโมเดล LSTM + MLP นั้นมีประสิทธิภาพและความเสถียรในการทำนายสูงกว่าการใช้โมเดล LSTM แบบพื้นฐาน โดยอ้างอิงจากการพิจารณาผ่านการใช้ Learning Curve

การพัฒนาโมเดลในการพยากรณ์สถานการณ์การแพร่ระบาดของโควิด-19 โดยใช้การเรียนรู้เชิงลึกและระบบสารสนเทศภูมิศาสตร์ ประกอบด้วยการใช้โมเดล LSTM และ MLP ร่วมกัน ผลการพัฒนาโมเดลสามารถทำนายจำนวนผู้ติดเชื้อสะสมและการเสียชีวิตสะสมได้อย่างแม่นยำ โดยมีค่าความแม่นยำในการทำนายจำนวนผู้ติดเชื้อสะสม และการทำนายจำนวนผู้เสียชีวิตสะสม การพัฒนาโมเดลนี้สอดคล้องกับวัตถุประสงค์ของงานวิจัยที่ตั้งไว้ดังนี้ วัตถุประสงค์ข้อที่ 1 การสร้างเครื่องมือสำหรับการวิเคราะห์สถานการณ์การแพร่ระบาดของเชื้อโควิด-19 โมเดลที่พัฒนาขึ้นสามารถวิเคราะห์

และคาดการณ์ได้อย่างมีประสิทธิภาพรวมไปถึงสามารถแสดงผลการทำนายบนแผนที่ของประเทศไทยได้ตามวัตถุประสงค์ที่ตั้งไว้ ในส่วนของวัตถุประสงค์ข้อที่ 2 การสร้างโมเดลสำหรับการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 โมเดล LSTM และ MLP ที่พัฒนาขึ้นมีความแม่นยำสูงในการทำนายจำนวนผู้ติดเชื้อสะสมและการเสียชีวิตสะสม ทำให้สามารถใช้โมเดลนี้เป็นโมเดลหลักในการทำนายได้

5.2 เว็บแอปพลิเคชัน

การสร้างเว็บแอปพลิเคชันสำหรับการแสดงผลการทำนายสถานการณ์การแพร่ระบาดของโควิด-19 บนแผนที่ของประเทศไทยเว็บแอปพลิเคชันนี้ได้รับการพัฒนาให้สามารถแสดงข้อมูลได้อย่างชัดเจนและเข้าถึงได้ง่าย แอปพลิเคชันนี้ออกแบบมาเพื่อให้ผู้ใช้สามารถดูข้อมูลทำนายได้สะดวก ผ่านการแสดงผลในรูปแบบของกราฟและแผนที่ ซึ่งช่วยให้สามารถติดตามและประเมินสถานการณ์การแพร่ระบาดในแต่ละจังหวัด

เว็บแอปพลิเคชันประกอบด้วยสามส่วนหลัก ได้แก่ ส่วนแผนที่ประเทศไทยที่แสดงผลการทำนายของแต่ละจังหวัดอย่างชัดเจน ผู้ใช้สามารถเลือกดูข้อมูลการติดเชื้อและการเสียชีวิตในแต่ละจังหวัดได้ทันที ส่วนกราฟแสดงสถิติที่แสดงข้อมูลการติดเชื้อสะสมและการเสียชีวิตสะสมในรูปแบบของกราฟเส้น ช่วยให้ผู้ใช้สามารถวิเคราะห์แนวโน้มของสถานการณ์ได้อย่างง่ายดาย และส่วนสุดท้ายคือส่วนกราฟแสดงการทำนายที่แสดงผลการทำนายจากโมเดล LSTM + MLP ทำให้ผู้ใช้สามารถดูการเปรียบเทียบระหว่างข้อมูลจริงและข้อมูลทำนายได้อย่างชัดเจน ทั้งสามส่วนนี้ทำให้เว็บแอปพลิเคชันเป็นเครื่องมือที่มีประสิทธิภาพในการติดตามและประเมินสถานการณ์การแพร่ระบาดของโควิด-19 การพัฒนาเว็บแอปพลิเคชันนี้สอดคล้องกับวัตถุประสงค์ข้อที่ 3 ดังนี้ วัตถุประสงค์ข้อที่ 3 การสร้างเว็บแอปพลิเคชันสำหรับการแสดงผลการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 บนแผนที่ของประเทศไทยเว็บแอปพลิเคชันที่พัฒนาขึ้นสามารถแสดงผลการทำนายได้อย่างมีประสิทธิภาพและใช้งานง่าย ช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลการทำนายและสถานการณ์การแพร่ระบาดได้

5.3 ข้อจำกัดในการใช้งาน

การพัฒนาโมเดล LSTM + MLP เพื่อใช้ในการทำนายสถานการณ์การแพร่ระบาดของเชื้อโควิด-19 ในครั้งนี้พบว่ายังมีข้อจำกัดที่เกี่ยวข้องของงานวิจัยนี้ทั้งหมด 2 ส่วนได้แก่ ข้อจำกัดด้านโมเดลและข้อจะกัดของเว็บแอปพลิเคชัน

1) ข้อจำกัดของโมเดล ข้อจำกัดของ AIP ของกรรมควบคุมโรคในปัจจุบันไม่ได้สามารถใช้งานได้เนื่องจากมีการเปลี่ยนที่อยู่ของ API และโครงสร้างของข้อมูลมีความเปลี่ยนไป เช่น ความถี่จากการรายงานสถานการณ์เปลี่ยนจากความถี่รายวันเป็นรายสัปดาห์จึงทำให้เกิดข้อจำกัดของโมเดลดังกล่าว

ในเรื่องของโครงสร้างของข้อมูลที่เปลี่ยนไปทำให้ต้องมีการปรับแต่งโมเดลเพื่อให้สอดคล้องกับโครงสร้างของข้อมูลที่เปลี่ยนไป

2) ข้อจะกัดของเว็บแอปพลิเคชัน เว็บแอปพลิเคชันที่พัฒนาขึ้นมานี้มีประสิทธิภาพในการแสดงผลการทำนายและช่วยให้สามารถติดตามสถานการณ์การแพร่ระบาดของโควิด-19 ได้อย่างมีประสิทธิภาพ แต่ก็ยังมีข้อจำกัดบางประการที่ต้องพิจารณา ข้อจำกัดแรกคือ ความถูกต้องของการทำนายที่ขึ้นอยู่กับคุณภาพและความสมบูรณ์ของข้อมูลที่ใช้ในการฝึกโมเดล รวมไปถึงความหลากหลายของข้อมูลที่เป็นปัจจัยในด้านต่างๆที่ส่งผลต่อการวิเคราะห์การแพร่ระบาดของเชื้อโควิด-19 ในอนาคต

5.4 ข้อเสนอแนะในการพัฒนาต่อ

การนำโมเดล LSTM + MLP เพื่อนำพัฒนาต่อยอดจากโมเดลปัจจุบันรวมไปถึงการพัฒนาเว็บแอปพลิเคชันนั้นสามารถนำไปพัฒนาต่อดังข้อเสนอแนะดังต่อไปนี้เพื่อเป็นแนวทางในการพัฒนาประสิทธิภาพของโมแดลและเว็บแอปพลิเคชัน

1) การพัฒนาโมเดล

แม้ว่าโมเดลและระบบทำนายสถานการณ์การแพร่ระบาดของโควิด-19 ในการศึกษานี้จะแสดงให้เห็นถึงประสิทธิภาพและความแม่นยำสูง แต่มีหลายประเด็นที่สามารถพัฒนาต่อไปได้ ประการแรก ควรเพิ่มการนำข้อมูลจากแหล่งอื่น ๆ เช่น ข้อมูลการเคลื่อนย้ายของประชากร ข้อมูลสิ่งแวดล้อม และข้อมูลการระบาดของโรคอื่น ๆ เพื่อเพิ่มความแม่นยำในการทำนาย การรวมข้อมูลจากหลายแหล่งจะช่วยให้โมเดลสามารถเรียนรู้และประมวลผลข้อมูลได้ครอบคลุมและลึกซึ้งมากขึ้น นอกจากนี้ ควรพิจารณาการนำระบบและโมเดลที่พัฒนาขึ้นนี้ไปประยุกต์ใช้กับโรคระบาดอื่น ๆ เช่น ไข้หวัดใหญ่ ไข้เลือดออก หรือโรคติดต่อทางเดินหายใจต่าง ๆ โดยการปรับแต่งโมเดลและข้อมูลที่ใช้ให้เหมาะสมกับลักษณะเฉพาะของแต่ละโรค การขยายการใช้งานในลักษณะนี้จะช่วยยกระดับความสามารถของระบบในการรับมือกับโรคระบาดในอนาคตและเพิ่มคุณค่าในการใช้งานของโมเดลทำนายนี้ นอกจากนี้ ยังสามารถใช้เป็นเครื่องมือในการวางแผนและบริหารจัดการสาธารณสุขในระดับประเทศอย่างมีประสิทธิภาพมากยิ่งขึ้น

จากการทดสอบและพัฒนาโมเดลในงานวิจัยนี้ พบว่าข้อมูลจาก API ของกรมควบคุมโรคมีการเปลี่ยนแปลงโครงสร้าง โดยเฉพาะอย่างยิ่งการเปลี่ยนความถี่ของการรายงานสถานการณ์โควิด-19 จากรายวันเป็นรายสัปดาห์ ซึ่งส่งผลกระทบต่อรูปแบบและการประมวลผลของข้อมูลในโมเดล หากต้องการพัฒนาโมเดลนี้ต่อในอนาคต จำเป็นจะต้องมีการปรับแต่งโมเดลในส่วนของการจัดการ

ข้อมูล เพื่อให้สามารถรองรับข้อมูลที่มีความถี่ในการรายงานที่แตกต่างกันได้อย่างมีประสิทธิภาพ การปรับแต่งดังกล่าวอาจรวมถึงการปรับพารามิเตอร์ของโมเดล

2) การพัฒนาเว็บแอปพลิเคชัน

เพื่อพัฒนาประสิทธิภาพและขยายขอบเขตการใช้งานของเว็บแอปพลิเคชันให้ครอบคลุมมากยิ่งขึ้น ข้อเสนอแนะสำคัญคือการเพิ่มข้อมูลสถิติต่างๆ ที่มีความเกี่ยวข้องกับสถานการณ์การแพร่ระบาด เช่น ข้อมูลการฉีดวัคซีนในแต่ละโดส รวมถึงข้อมูลความหนาแน่นของประชากรในแต่ละจังหวัด การเพิ่มข้อมูลเหล่านี้จะช่วยให้การวิเคราะห์สถานการณ์มีความละเอียดและครบถ้วนมากยิ่งขึ้น นอกจากนี้ การออกแบบเว็บแอปพลิเคชันให้มีลักษณะคล้ายกับแดชบอร์ด (Dashboard) มากขึ้น จะช่วยให้ผู้ใช้งานสามารถดูข้อมูลสถิติต่างๆ ได้ในที่เดียวอย่างสะดวกและรวดเร็ว ทั้งนี้ การแสดงผลข้อมูลแบบแดชบอร์ดจะช่วยให้ผู้ใช้สามารถมองเห็นภาพรวมของสถานการณ์ได้อย่างชัดเจนและสามารถปรับใช้ข้อมูลเพื่อการวิเคราะห์และตัดสินใจได้อย่างมีประสิทธิภาพยิ่งขึ้น

ภาคผนวก

ตาราง 17 ผลการทำนายของโมเดล LSTM แบบพื้นฐานจากชุดมูลการติดเชื้อสะสม

File Name	RMSE	MAE	MAPE	MSE
Amnat Charoen	351.42	351.41	0.03	123497.78
Ang Thong	2270.25	2270.23	0.08	5154050.10
Bangkok	145212.37	145211.03	0.15	21086633060.00
Bueng Kan	1062.62	1062.61	0.08	1129159.14
Buriram	6675.49	6674.88	0.10	44562178.68
Chachoengsao	5926.37	5926.31	0.07	35121867.43
Chainat	457.65	457.26	0.06	209443.26
Chaiyaphum	2273.50	2273.44	0.07	5168814.25
Chanthaburi	3265.57	3265.32	0.07	10663971.54
Chiang Mai	4386.69	4386.68	0.06	19243068.25
Chiang Rai	1557.60	1557.53	0.13	2426130.68
Chonburi	12593.87	12593.35	0.05	158605445.30
Chumphon	3024.61	3023.67	0.09	9148276.78
Kalasin	2285.23	2285.05	0.06	5222280.21
Kamphaeng Phet	620.15	619.94	0.02	384580.88
Kanchanaburi	4869.63	4869.62	0.09	23713335.57
Khon Kaen	2720.19	2719.42	0.03	7399424.40
Krabi	2048.55	2048.54	0.08	4196546.50
Lampang	1283.36	1283.15	0.09	1647000.69
Lamphun	710.32	710.27	0.10	504556.18
Loei	795.39	794.75	0.04	632644.27
Lopburi	2594.13	2593.95	0.07	6729525.98
Mae Hong Son	171.23	170.92	0.03	29318.98
Maha Sarakham	1102.44	1102.42	0.03	1215372.93
Mukdahan	532.21	532.20	0.06	283246.97
Nakhon Nayok	1674.32	1674.21	0.07	2803340.24
Nakhon Pathom	6127.80	6127.73	0.07	37549897.89
Nakhon Phanom	102.66	102.00	0.01	10540.03
Nakhon Ratchasima	8068.89	8068.18	0.09	65107038.75

Nakhon Sawan	3761.82	3761.60	0.09	14151315.26
Nakhon Si Thammarat	8933.68	8933.55	0.07	79810720.40
Nan	870.14	867.95	0.06	757144.61
Narathiwat	3859.62	3859.62	0.08	14896653.38
Nong Bua Lam Phu	37.71	35.18	0.00	1421.86
Nong Khai	4420.03	4420.00	0.16	19536706.37
Nonthaburi	8899.17	8898.64	0.06	79195214.49
Pathum Thani	7891.07	7890.95	0.09	62268967.95
Pattani	1397.82	1397.81	0.02	1953894.59
Phang Nga	543.04	543.04	0.03	294890.54
Phatthalung	2186.02	2186.02	0.06	4778684.61
Phayao	680.63	680.54	0.07	463253.08
Phetchabun	1694.54	1694.42	0.07	2871459.00
Phetchaburi	1936.02	1935.92	0.04	3748179.71
Phichit	772.80	772.72	0.07	597218.51
Phitsanulok	1684.84	1684.65	0.05	2838700.91
Phra Nakhon Si Ayutthaya	2092.40	2092.31	0.03	4378122.23
Phrae	260.67	260.22	0.02	67947.46
Phuket	3265.34	3265.34	0.05	10662477.44
Prachin Buri	2739.97	2739.88	0.05	7507437.19
Prachuap Khiri Khan	3708.80	3708.63	0.08	13755177.37
Ranong	737.95	737.95	0.04	544573.87
Ratchaburi	2578.50	2578.33	0.03	6648663.32
Rayong	7764.21	7764.15	0.09	60283005.72
Roi Et	5154.38	5154.22	0.09	26567649.73
Sa Kaeo	3813.84	3813.53	0.10	14545377.88
Sakon Nakhon	944.93	944.83	0.03	892896.01
Samut Prakan	10357.16	10356.95	0.04	107270814.90
Samut Sakhon	5379.36	5379.00	0.03	28937560.01
Samut Songkhram	867.19	867.12	0.03	752026.00
Saraburi	2037.11	2037.03	0.04	4149797.29
Satun	1463.67	1463.66	0.07	2142332.86
Sing Buri	1651.16	1651.11	0.14	2726324.79
Sisaket	1805.31	1805.23	0.04	3259143.18

Songkhla	8007.23	8007.20	0.08	64115748.49
Sukhothai	1171.09	1170.91	0.05	1371460.12
Suphan Buri	7284.49	7284.33	0.14	53063742.55
Surat Thani	2293.05	2292.68	0.05	5258090.67
Surin	3748.89	3747.63	0.07	14054192.83
Tak	2057.37	2057.31	0.05	4232762.11
Trang	364.72	364.53	0.01	133018.49
Trat	2144.75	2144.66	0.11	4599956.43
Ubon Ratchathani	5685.41	5684.83	0.09	32323937.51
Udon Thani	3655.65	3655.59	0.07	13363769.24
Uthai Thani	1353.45	1353.19	0.09	1831838.32
Uttaradit	932.41	932.33	0.07	869384.36
Yala	1030.23	1030.18	0.02	1061365.97
Yasothon	1077.12	1077.09	0.05	1160188.42

ตาราง 18 ผลการทำนายของโมเดล LSTM แบบพื้นฐานจากชุดมูลการเสียชีวิตสะสม

File Name	MSE	RMSE	MAE	MAPE
Amnat Charoen	20.10	4.48	4.47	0.07
Ang Thong	428.31	20.70	20.68	0.09
Bangkok	445742.60	667.64	667.54	0.08
Bueng Kan	69.29	8.32	8.32	0.21
Buriram	25.62	5.06	5.05	0.03
Chachoengsao	2065.17	45.44	45.37	0.10
Chainat	0.10	0.32	0.26	0.01
Chaiyaphum	96.47	9.82	9.80	0.05
Chanthaburi	93.12	9.65	9.16	0.04
Chiang Mai	196882.33	443.71	443.71	0.88
Chiang Rai	254.95	15.97	15.97	0.07
Chonburi	888.66	29.81	29.45	0.02
Chumphon	11.97	3.46	3.44	0.02
Kalasin	1517.65	38.96	38.91	0.16
Kamphaeng Phet	500.11	22.36	22.36	0.09
Kanchanaburi	3074.90	55.45	55.44	0.16
Khon Kaen	28740.84	169.53	169.52	0.59
Krabi	554.25	23.54	23.51	0.12
Lampang	321.99	17.94	17.77	0.11
Lamphun	289.19	17.01	16.82	0.24
Loei	11.70	3.42	2.92	0.02
Lopburi	874.03	29.56	29.56	0.06
Mae Hong Son	318.70	17.85	17.81	0.22
Maha Sarakham	402.44	20.06	20.06	0.13
Mukdahan	173.85	13.19	13.19	0.20
Nakhon Nayok	1766.61	42.03	42.03	0.17
Nakhon Pathom	528.79	23.00	22.99	0.03
Nakhon Phanom	878.35	29.64	29.63	0.27
Nakhon Ratchasima	240877.29	490.79	490.79	0.74
Nakhon Sawan	9365.39	96.77	96.50	0.23
Nakhon Si Thammarat	8.33	2.89	2.86	0.00
Nan	10.83	3.29	3.29	0.07

Narathiwat	226.03	15.03	15.03	0.04
Nong Bua Lam Phu	20.03	4.48	4.13	0.03
Nong Khai	43.92	6.63	6.63	0.06
Nonthaburi	8085.56	89.92	89.91	0.17
Pathum Thani	1549.87	39.37	39.36	0.04
Pattani	528.14	22.98	22.98	0.04
Phang Nga	9.87	3.14	3.14	0.05
Phatthalung	1.68	1.30	1.30	0.01
Phayao	842.89	29.03	28.91	0.38
Phetchabun	3.85	1.96	1.96	0.01
Phetchaburi	623.76	24.98	24.85	0.09
Phichit	0.92	0.96	0.86	0.01
Phitsanulok	2217.16	47.09	47.08	0.26
Phra Nakhon Si Ayutthaya	3469.68	58.90	58.89	0.11
Phrae	640.94	25.32	24.91	0.23
Phuket	128.33	11.33	11.23	0.04
Prachin Buri	352.83	18.78	18.76	0.06
Prachuap Khiri Khan	149.36	12.22	12.20	0.05
Ranong	33.80	5.81	5.81	0.04
Ratchaburi	20.59	4.54	4.40	0.01
Rayong	145.57	12.07	11.92	0.02
Roi Et	86.58	9.30	9.30	0.03
Sa Kaeo	8.51	2.92	2.91	0.02
Sakon Nakhon	875.72	29.59	29.59	0.16
Samut Prakan	28273.84	168.15	168.11	0.09
Samut Sakhon	651.52	25.52	25.51	0.02
Samut Songkhram	152.66	12.36	12.36	0.06
Saraburi	1202.41	34.68	34.67	0.06
Satun	225.79	15.03	15.03	0.08
Sing Buri	3925.43	62.65	62.64	0.40
Sisaket	2329.13	48.26	48.23	0.15
Songkhla	1689.40	41.10	41.10	0.09
Sukhothai	547.99	23.41	23.30	0.09
Suphan Buri	108.59	10.42	10.20	0.02

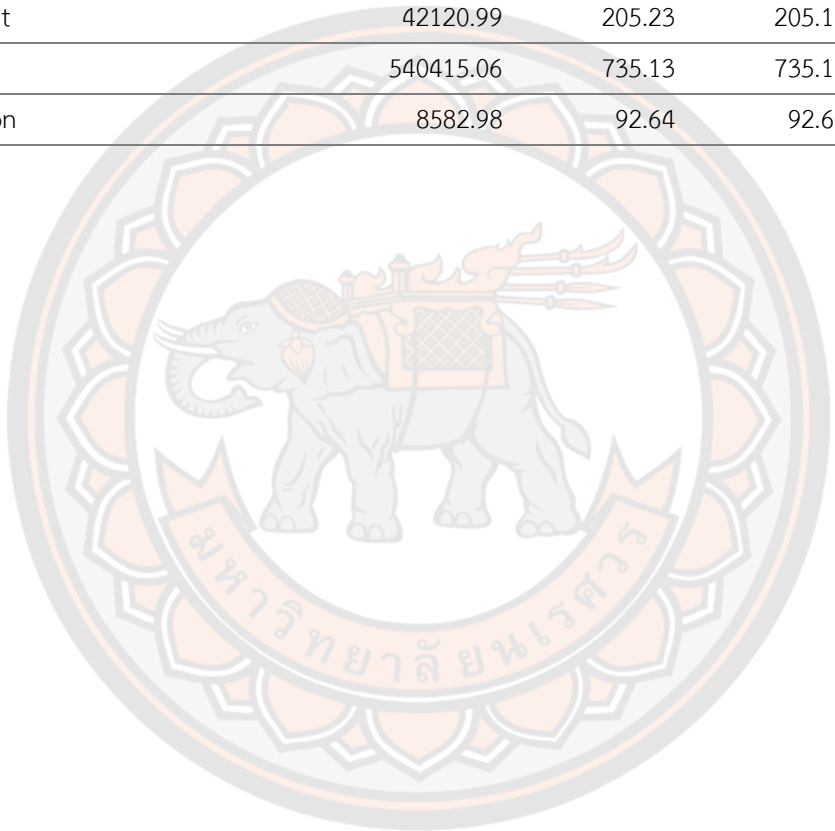
Surat Thani	375.07	19.37	19.14	0.06
Surin	3180.79	56.40	56.18	0.26
Tak	286.19	16.92	16.77	0.04
Trang	9.69	3.11	3.05	0.02
Trat	9.69	3.11	3.01	0.01
Ubon Ratchathani	366.98	19.16	19.15	0.04
Udon Thani	4.60	2.14	2.03	0.01
Uthai Thani	6698.53	81.84	81.84	0.50
Uttaradit	5.50	2.34	2.34	0.02
Yala	381.28	19.53	19.52	0.05
Yasothon	422.65	20.56	20.53	0.13

ตาราง 19 ผลการทำนายของโมเดล LSTM + MLP จากชุดมูลการติดเชื้อสะสม

File Name	MSE	RMSE	MAE	MAPE
Amnat Charoen	837.48	28.94	28.90	0.00
Ang Thong	22156.95	148.85	148.82	0.01
Bangkok	2425294.84	1557.34	1514.36	0.00
Bueng Kan	1501.63	38.75	38.72	0.00
Buriram	36414.68	190.83	189.86	0.00
Chachoengsao	63191.06	251.38	250.85	0.00
Chainat	119.60	10.94	10.34	0.00
Chaiyaphum	2784.69	52.77	52.16	0.00
Chanthaburi	68699.98	262.11	261.65	0.01
Chiang Mai	469821.31	685.44	685.42	0.01
Chiang Rai	5576.80	74.68	74.30	0.01
Chonburi	35791.07	189.19	175.53	0.00
Chumphon	31153.18	176.50	173.58	0.01
Kalasin	2781.36	52.74	52.47	0.00
Kamphaeng Phet	7642.27	87.42	87.18	0.00
Kanchanaburi	2138.26	46.24	46.09	0.00
Khon Kaen	621.75	24.93	22.08	0.00
Krabi	70788.55	266.06	264.69	0.01
Lampang	15268.63	123.57	123.30	0.01
Lamphun	104852.62	323.81	323.34	0.05
Loei	53705.22	231.74	231.59	0.01
Lopburi	645.81	25.41	24.42	0.00
Mae Hong Son	668.25	25.85	25.57	0.00
Maha Sarakham	122.63	11.07	8.74	0.00
Mukdahan	399.26	19.98	19.95	0.00
Nakhon Nayok	799.03	28.27	27.91	0.00
Nakhon Pathom	109552.45	330.99	330.80	0.00
Nakhon Phanom	4239.76	65.11	64.74	0.00
Nakhon Ratchasima	198259.20	445.26	443.04	0.01
Nakhon Sawan	24618.20	156.90	156.39	0.00
Nakhon Si Thammarat	8787.45	93.74	90.49	0.00
Nan	16293.44	127.65	126.73	0.01

Narathiwat	453555.27	673.47	673.46	0.01
Nong Bua Lam Phu	25.85	5.08	4.26	0.00
Nong Khai	7475.77	86.46	86.34	0.00
Nonthaburi	82418.10	287.09	283.68	0.00
Pathum Thani	168113.37	410.02	408.53	0.00
Pattani	65889.33	256.69	256.68	0.00
Phang Nga	4178.92	64.64	64.64	0.00
Phatthalung	429.65	20.73	20.67	0.00
Phayao	2909.28	53.94	52.80	0.01
Phetchabun	2524.59	50.25	48.83	0.00
Phetchaburi	32949.48	181.52	181.38	0.00
Phichit	687.94	26.23	24.84	0.00
Phitsanulok	37067.84	192.53	192.23	0.01
Phra Nakhon Si Ayutthaya	17015.95	130.45	130.25	0.00
Phrae	9892.01	99.46	99.19	0.01
Phuket	102028.36	319.42	319.28	0.00
Prachin Buri	115176.97	339.38	339.31	0.01
Prachuap Khiri Khan	1808.40	42.53	39.85	0.00
Ranong	10276.52	101.37	101.35	0.00
Ratchaburi	170931.34	413.44	413.31	0.00
Rayong	116939.06	341.96	341.77	0.00
Roi Et	8431.46	91.82	91.52	0.00
Sa Kaeo	2336.27	48.33	42.79	0.00
Sakon Nakhon	2427.76	49.27	49.18	0.00
Samut Prakan	1085631.45	1041.94	1041.11	0.00
Samut Sakhon	705182.75	839.75	838.57	0.00
Samut Songkhram	6365.07	79.78	79.75	0.00
Saraburi	379623.35	616.14	615.98	0.01
Satun	5429.21	73.68	73.56	0.00
Sing Buri	460.29	21.45	21.31	0.00
Sisaket	14595.31	120.81	120.64	0.00
Songkhla	1203426.86	1097.01	1096.36	0.01
Sukhothai	503.79	22.45	20.42	0.00
Suphan Buri	20384.07	142.77	142.45	0.00

Surat Thani	8369.32	91.48	89.35	0.00
Surin	30486.43	174.60	172.74	0.00
Tak	411453.10	641.45	640.53	0.02
Trang	166872.74	408.50	407.85	0.01
Trat	14704.83	121.26	120.87	0.01
Ubon Ratchathani	15354.81	123.91	122.42	0.00
Udon Thani	179290.62	423.43	423.39	0.01
Uthai Thani	7781.67	88.21	87.40	0.01
Uttaradit	42120.99	205.23	205.17	0.02
Yala	540415.06	735.13	735.11	0.01
Yasothon	8582.98	92.64	92.61	0.00

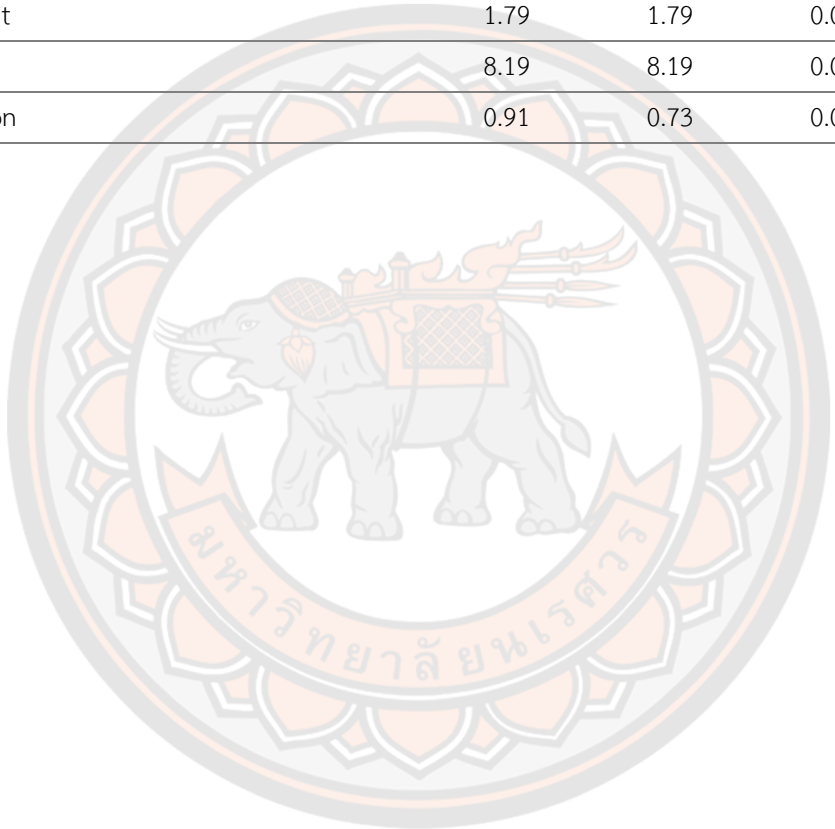


ตาราง 20 ผลการทำนายของโมเดล LSTM + MLP จากชุดมูลการเสียชีวิตสะสมโดย

File Name	RMSE	MAE	MAPE	MSE
Amnat Charoen	0.63	0.58	0.01	0.40
Ang Thong	5.27	5.20	0.02	27.76
Bangkok	92.29	92.13	0.01	8517.19
Bueng Kan	0.37	0.37	0.01	0.13
Buriram	0.59	0.43	0.00	0.35
Chachoengsao	7.84	7.81	0.02	61.47
Chainat	0.66	0.55	0.01	0.44
Chaiyaphum	2.69	2.64	0.01	7.24
Chanthaburi	7.77	7.65	0.04	60.35
Chiang Mai	9.65	9.57	0.02	93.13
Chiang Rai	8.18	8.18	0.04	66.85
Chonburi	14.17	13.76	0.01	200.93
Chumphon	4.40	4.36	0.02	19.35
Kalasin	1.65	1.44	0.01	2.74
Kamphaeng Phet	2.80	2.75	0.01	7.84
Kanchanaburi	4.77	4.68	0.01	22.73
Khon Kaen	3.39	3.26	0.01	11.51
Krabi	4.10	4.00	0.02	16.82
Lampang	8.05	7.93	0.05	64.87
Lamphun	3.18	3.09	0.04	10.13
Loei	3.32	3.02	0.02	11.03
Lopburi	2.19	2.17	0.00	4.80
Mae Hong Son	2.60	2.58	0.03	6.73
Maha Sarakham	2.32	2.32	0.01	5.39
Mukdahan	1.20	1.20	0.02	1.44
Nakhon Nayok	1.92	1.66	0.01	3.67
Nakhon Pathom	24.77	24.77	0.03	613.62
Nakhon Phanom	1.33	1.19	0.01	1.78
Nakhon Ratchasima	9.16	9.08	0.01	83.91
Nakhon Sawan	22.85	22.81	0.05	522.10
Nakhon Si Thammarat	1.69	1.62	0.00	2.87
Nan	0.27	0.27	0.01	0.07

Narathiwat	2.59	2.59	0.01	6.73
Nong Bua Lam Phu	3.82	3.73	0.03	14.59
Nong Khai	1.82	1.82	0.02	3.32
Nonthaburi	24.79	24.78	0.05	614.31
Pathum Thani	28.34	28.34	0.03	803.32
Pattani	3.20	3.17	0.01	10.21
Phang Nga	0.72	0.72	0.01	0.52
Phatthalung	1.99	1.99	0.01	3.94
Phayao	3.89	3.79	0.05	15.12
Phetchabun	5.46	5.46	0.03	29.87
Phetchaburi	9.72	9.67	0.04	94.56
Phichit	0.73	0.64	0.00	0.53
Phitsanulok	1.07	1.04	0.01	1.14
Phra Nakhon Si Ayutthaya	15.73	15.72	0.03	247.52
Phrae	4.28	3.88	0.04	18.28
Phuket	6.47	6.41	0.02	41.84
Prachin Buri	7.20	7.16	0.02	51.79
Prachuap Khiri Khan	0.44	0.35	0.00	0.19
Ranong	3.52	3.52	0.03	12.42
Ratchaburi	13.00	12.96	0.03	169.01
Rayong	11.57	11.41	0.02	133.90
Roi Et	3.88	3.88	0.01	15.05
Sa Kaeo	0.24	0.15	0.00	0.06
Sakon Nakhon	2.30	2.30	0.01	5.29
Samut Prakan	23.80	23.73	0.01	566.31
Samut Sakhon	0.92	0.81	0.00	0.84
Samut Songkhram	0.19	0.19	0.00	0.04
Saraburi	11.65	11.63	0.02	135.65
Satun	2.01	2.01	0.01	4.03
Sing Buri	0.84	0.73	0.00	0.71
Sisaket	1.44	0.97	0.00	2.08
Songkhla	2.25	2.24	0.00	5.08
Sukhothai	1.99	1.65	0.01	3.96
Suphan Buri	2.03	1.52	0.00	4.12

Surat Thani	14.12	13.92	0.04	199.28
Surin	8.03	7.70	0.04	64.40
Tak	2.99	2.86	0.01	8.93
Trang	3.58	3.55	0.02	12.79
Trat	2.86	2.71	0.01	8.16
Ubon Ratchathani	7.33	7.33	0.01	53.79
Udon Thani	7.56	7.55	0.02	57.22
Uthai Thani	1.47	1.26	0.01	2.15
Uttaradit	1.79	1.79	0.01	3.20
Yala	8.19	8.19	0.02	67.08
Yasothon	0.91	0.73	0.00	0.83



บรรณานุกรม

Uncategorized References

1. Zeng, J.-H., et al., *First case of COVID-19 complicated with fulminant myocarditis: a case report and insights.* Infection, 2020. **48**: p. 773-777.
2. Strugnell, R. and N. Wang, *Sustained neutralising antibodies in the Wuhan population suggest durable protection against SARS-CoV-2.* The Lancet, 2021. **397**(10279): p. 1037-1039.
3. Li, Y.C., W.Z. Bai, and T. Hashikawa, *The neuroinvasive potential of SARS‐CoV2 may play a role in the respiratory failure of COVID‐19 patients.* Journal of medical virology, 2020. **92**(6): p. 552-555.
4. !!! INVALID CITATION !!! [4].
5. Organization, W.H. *WHO Coronavirus (COVID-19) Dashboard*. 2023 [cited 2022 2022]; Available from: https://covid19.who.int/
6. Tantrakarnapa, K., B. Bhopdhornangkul, and K. Nakhaapakorn, *Influencing factors of COVID-19 spreading: a case study of Thailand.* Journal of Public Health, 2020: p. 1-7.
7. Leerapan, B., et al., *How systems respond to policies: intended and unintended consequences of COVID-19 lockdown policies in Thailand.* Health Policy and Planning, 2022. **37**(2): p. 292-293.
8. Pornpatanapaisansakul, K. *Labor Market Digital Transformation 2020*. 2020 [cited 2022 2022]; Available from: https://www.bot.or.th/Thai/MonetaryPolicy/ArticleAndResearch/Pages/FAQ171.
9. Anusonphat, N. and C. Poompurk, ทาง รอด และ แนวทาง การ ปรับ ตัว ของ วิสาหกิจ ชุมชน แนว วิถี ใหม่ หลัง วิกฤติ *COVID-19* ของ ไทย. Journal of MCU Nakhondhat, 2022. **9**(1): p. 1-19.
10. Thailand, M.o.P.H. *The report of COVID-19 situation in Thailand*. 2022 [cited 2022 2022]; Available from: https://ddc.moph.go.th/covid19-dashboard/.
11. Montaha, S., et al., *Timedistributed-cnn-lstm: A hybrid approach combining cnn and lstm to classify brain tumor on 3d mri scans performing ablation study.*

IEEE Access, 2022. **10**: p. 60039-60059.

12. Cheng, Z., Y. Zhang, and C. Tang, *Solving Monocular Sensors Depth Prediction Using MLP-Based Architecture and Multi-Scale Inverse Attention.* IEEE Sensors Journal, 2022. **22**(16): p. 16178-16189.
13. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.
14. Lawrence, J., *Introduction to neural networks*. 1993: California Scientific Software.
15. O'shea, T. and J. Hoydis, *An introduction to deep learning for the physical layer.* IEEE Transactions on Cognitive Communications and Networking, 2017. **3**(4): p. 563-575.
16. Organization, W.H., *Coronavirus disease 2019 (COVID-19): situation report, 73.* 2020.
17. Rusk, N., *Deep learning.* Nature Methods, 2016. **13**(1): p. 35-35.
18. Gardner, M.W. and S. Dorling, *Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences.* Atmospheric environment, 1998. **32**(14-15): p. 2627-2636.
19. Li, F., et al., *Input layer regularization of multilayer feedforward neural networks.* IEEE Access, 2017. **5**: p. 10979-10985.
20. Huang, G.-B., Y.-Q. Chen, and H.A. Babri, *Classification ability of single hidden layer feedforward neural networks.* IEEE transactions on neural networks, 2000. **11**(3): p. 799-801.
21. Le, H.-S., et al., *Structured output layer neural network language models for speech recognition.* IEEE Transactions on Audio, Speech, and Language Processing, 2012. **21**(1): p. 197-206.
22. Medsker, L.R. and L. Jain, *Recurrent neural networks.* Design and Applications, 2001. **5**(64-67): p. 2.
23. Schuster, M. and K.K. Paliwal, *Bidirectional recurrent neural networks.* IEEE transactions on Signal Processing, 1997. **45**(11): p. 2673-2681.
24. Cambria, E. and B. White, *Jumping NLP curves: A review of natural language processing research.* IEEE Computational intelligence magazine, 2014. **9**(2): p. 48-

57.

25. Singh, S.P., et al. *Machine translation using deep learning: An overview*. in *2017 international conference on computer, communications and electronics (comptelix)*. 2017. IEEE.
26. Kousik, N., et al., *Improved salient object detection using hybrid Convolution Recurrent Neural Network.* Expert Systems with Applications, 2021. **166**: p. 114064.
27. Abdel-Nasser, M. and K. Mahmoud, *Accurate photovoltaic power forecasting models using deep LSTM-RNN.* Neural computing and applications, 2019. **31**: p. 2727-2740.
28. Shi, H., M. Xu, and R. Li, *Deep learning for household load forecasting—A novel pooling deep RNN.* IEEE Transactions on Smart Grid, 2017. **9**(5): p. 5271-5280.
29. Tsoi, A.C., *Recurrent neural network architectures: an overview.* International School on Neural Networks, Initiated by IIASS and EMFCSC, 1997: p. 1-26.
30. Yu, Y., et al., *A review of recurrent neural networks: LSTM cells and network architectures.* Neural computation, 2019. **31**(7): p. 1235-1270.
31. Yunpeng, L., et al. *Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network*. in *2017 14th web information systems and applications conference (WISA)*. 2017. IEEE.
32. Onan, A., *Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification.* Journal of King Saud University-Computer and Information Sciences, 2022. **34**(5): p. 2098-2117.
33. Fu, Y., et al. *An intelligent network attack detection method based on rnn*. in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. 2018. IEEE.
34. Brillinger, D.R., *Time series: data analysis and theory*. 2001: SIAM.
35. !!! INVALID CITATION !!! [37-39].
36. Mao, S. and E. Sejdić, *A review of recurrent neural network-based methods in computational physiology.* IEEE Transactions on Neural Networks and Learning Systems, 2022.

37. Strobelt, H., et al., *Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks.* IEEE transactions on visualization and computer graphics, 2017. **24**(1): p. 667-676.
38. He, Y. and S. Young, *Semantic processing using the hidden vector state model.* Computer speech & language, 2005. **19**(1): p. 85-106.
39. Hori, C., et al. *Driver confusion status detection using recurrent neural networks*. in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. 2016. IEEE.
40. Lin, L.-J. and T.M. Mitchell, *Reinforcement learning with hidden states.* From animals to animats, 1993. **2**: p. 271-280.
41. McCallum, R.A., *Instance-based utile distinctions for reinforcement learning with hidden state*, in *Machine Learning Proceedings 1995*. 1995, Elsevier. p. 387-395.
42. Chung, J., et al., *A recurrent latent variable model for sequential data.* Advances in neural information processing systems, 2015. **28**.
43. Wang, J., *A recurrent neural network for real-time matrix inversion.* Applied Mathematics and Computation, 1993. **55**(1): p. 89-100.
44. Choi, H., *Persistent hidden states and nonlinear transformation for long short-term memory.* Neurocomputing, 2019. **331**: p. 458-464.
45. Liu, W., et al., *THAT-Net: Two-layer hidden state aggregation based two-stream network for traffic accident prediction.* Information Sciences, 2023. **634**: p. 744-760.
46. Hernando, D., V. Crespi, and G. Cybenko, *Efficient computation of the hidden Markov model entropy for a given observation sequence.* IEEE transactions on information theory, 2005. **51**(7): p. 2681-2685.
47. Ming, Y., et al. *Understanding hidden memories of recurrent neural networks*. in *2017 IEEE conference on visual analytics science and technology (VAST)*. 2017. IEEE.
48. Titos, M., et al., *Toward Knowledge Extraction in Classification of Volcano-Seismic Events: Visualizing Hidden States in Recurrent Neural Networks.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022. **15**: p. 2311-2325.

49. Zimmer, C., *Reconstructing the hidden states in time course data of stochastic models.* Mathematical BioSciences, 2015. **269**: p. 117-129.
50. Chatzikonstantinou, C., et al., *Recurrent neural network pruning using dynamical systems and iterative fine-tuning.* Neural Networks, 2021. **143**: p. 475-488.
51. Khalifa, Y., D. Mandic, and E. Sejdić, *A review of Hidden Markov models and Recurrent Neural Networks for event detection and localization in biomedical signals.* Information Fusion, 2021. **69**: p. 52-72.
52. Hsu, W.-N., Y. Zhang, and J. Glass. *A prioritized grid long short-term memory RNN for speech recognition*. in *2016 IEEE Spoken Language Technology Workshop (SLT).* 2016. IEEE.
53. Zhang, D., et al., *Landslide risk prediction model using an attention-based temporal convolutional network connected to a recurrent neural network.* IEEE Access, 2022. **10**: p. 37635-37645.
54. Zhang, T., et al., *Spatial–temporal recurrent neural network for emotion recognition.* IEEE transactions on cybernetics, 2018. **49**(3): p. 839-847.
55. Sutskever, I. and G. Hinton, *Temporal-kernel recurrent neural networks.* Neural Networks, 2010. **23**(2): p. 239-243.
56. Greff, K., et al., *LSTM: A search space odyssey.* IEEE transactions on neural networks and learning systems, 2016. **28**(10): p. 2222-2232.
57. Son, H.H., *Toward a proposed framework for mood recognition using LSTM Recurrent Neuron Network.* Procedia computer science, 2017. **109**: p. 1028-1034.
58. Xue, K., et al., *An improved generic hybrid prognostic method for RUL prediction based on PF-LSTM learning.* IEEE Transactions on Instrumentation and Measurement, 2023. **72**: p. 1-21.
59. Tasdelen, A. and B. Sen, *A hybrid CNN-LSTM model for pre-miRNA classification.* Scientific reports, 2021. **11**(1): p. 14125.
60. Archith, S., et al., *Analysis of M-SEIR and LSTM Models for the Prediction of COVID-19 Using RMSLE.* Fundamentals and Methods of Machine and Deep Learning: Algorithms, Tools and Applications, 2022: p. 101-119.
61. Chimmula, V.K.R. and L. Zhang, *Time series forecasting of COVID-19 transmission*

in Canada using LSTM networks. Chaos, solitons & fractals, 2020. **135**: p. 109864.

62. Smagulova, K. and A.P. James, *A survey on LSTM memristive neural network architectures and applications.* The European Physical Journal Special Topics, 2019. **228**(10): p. 2313-2324.
63. Khan, L., et al., *Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media.* Applied Sciences, 2022. **12**(5): p. 2694.
64. Lindemann, B., et al., *A survey on long short-term memory networks for time series prediction.* Procedia CIRP, 2021. **99**: p. 650-655.
65. Wu, B., et al., *GBC: An energy-efficient LSTM accelerator with gating units level balanced compression strategy.* IEEE Transactions on Circuits and Systems I: Regular Papers, 2022. **69**(9): p. 3655-3665.
66. ArunKumar, K., et al., *Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends.* Alexandria engineering journal, 2022. **61**(10): p. 7585-7603.
67. Du, J., C.-M. Vong, and C.P. Chen, *Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification.* IEEE transactions on cybernetics, 2020. **51**(3): p. 1586-1597.
68. Dubey, S.R., S.K. Singh, and B.B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark.* Neurocomputing, 2022.
69. Farzad, A., H. Mashayekhi, and H. Hassanpour, *A comparative performance analysis of different activation functions in LSTM networks for classification.* Neural Computing and Applications, 2019. **31**: p. 2507-2521.
70. Jalali, S.M.J., et al., *Automated deep CNN-LSTM architecture design for solar irradiance forecasting.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021. **52**(1): p. 54-65.
71. Graves, A. and A. Graves, *Long short-term memory.* Supervised sequence

labelling with recurrent neural networks, 2012: p. 37-45.

72. Ebrahimi, A., et al., *Deep sequence modelling for Alzheimer's disease detection using MRI.* Computers in Biology and Medicine, 2021. **134**: p. 104537.

73. Khalil, K., et al., *Economic LSTM approach for recurrent neural networks.* IEEE Transactions on Circuits and Systems II: Express Briefs, 2019. **66**(11): p. 1885-1889.

74. Quan, R., et al., *Holistic LSTM for pedestrian trajectory prediction.* IEEE transactions on image processing, 2021. **30**: p. 3229-3239.

75. Wu, J.M.-T., et al., *Convert index trading to option strategies via LSTM architecture.* Neural Computing and Applications, 2020: p. 1-18.

76. Jiang, B., et al., *Attention-LSTM architecture combined with Bayesian hyperparameter optimization for indoor temperature prediction.* Building and Environment, 2022. **224**: p. 109536.

77. Agga, A., et al., *Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models.* Renewable Energy, 2021. **177**: p. 101-112.

78. Zhang, X., et al. *At-lstm: An attention-based lstm model for financial time series prediction*. in *IOP Conference Series: Materials Science and Engineering*. 2019. IOP Publishing.

79. Gers, F.A., N.N. Schraudolph, and J. Schmidhuber, *Learning precise timing with LSTM recurrent networks.* Journal of machine learning research, 2002. **3**(Aug): p. 115-143.

80. Zhou, Y. and X. Jiao, *Intelligent analysis system for signal processing tasks based on LSTM recurrent neural network algorithm.* Neural Computing and Applications, 2022. **34**(15): p. 12257-12269.

81. Gers, F.A. and E. Schmidhuber, *LSTM recurrent networks learn simple context-free and context-sensitive languages.* IEEE transactions on neural networks, 2001. **12**(6): p. 1333-1340.

82. Rahman, L., N. Mohammed, and A.K. Al Azad. *A new LSTM model by introducing biological cell state*. in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*. 2016. IEEE.

83. DiPietro, R. and G.D. Hager, *Deep learning: RNNs and LSTM*, in *Handbook of medical image computing and computer assisted intervention*. 2020, Elsevier. p. 503-519.

84. Guo, T., T. Lin, and N. Antulov-Fantulin. *Exploring interpretable LSTM neural networks over multi-variable data*. in *International conference on machine learning*. 2019. PMLR.

85. Ramchoun, H., et al., *Multilayer perceptron: Architecture optimization and training.* 2016.

86. Jain, A.K., J. Mao, and K.M. Mohiuddin, *Artificial neural networks: A tutorial.* Computer, 1996. **29**(3): p. 31-44.

87. Nie, S., M. Zheng, and Q. Ji, *The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision.* IEEE Signal Processing Magazine, 2018. **35**(1): p. 101-111.

88. Lathuilière, S., et al., *A comprehensive analysis of deep regression.* IEEE transactions on pattern analysis and machine intelligence, 2019. **42**(9): p. 2065-2081.

89. Palo, H.K., M. Chandra, and M.N. Mohanty, *Emotion recognition using MLP and GMM for Oriya language.* International Journal of Computational Vision and Robotics, 2017. **7**(4): p. 426-442.

90. Trenn, S., *Multilayer perceptrons: Approximation order and necessary number of hidden units.* IEEE transactions on neural networks, 2008. **19**(5): p. 836-844.

91. Karsoliya, S., *Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture.* International Journal of Engineering Trends and Technology, 2012. **3**(6): p. 714-717.

92. Taud, H. and J. Mas, *Multilayer perceptron (MLP).* Geomatic approaches for modeling land change scenarios, 2018: p. 451-455.

93. Panchal, G., et al., *Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers.* International Journal of Computer Theory and Engineering, 2011. **3**(2): p. 332-337.

94. Amid, S. and T. Mesri Gundoshmian, *Prediction of output energies for broiler production using linear regression, ANN (MLP, RBF), and ANFIS models.*

Environmental Progress & Sustainable Energy, 2017. **36**(2): p. 577-585.

95. Lillicrap, T.P., et al., *Backpropagation and the brain.* Nature Reviews Neuroscience, 2020. **21**(6): p. 335-346.

96. Pourghasemi, H.R., et al., *Spatial modeling, risk mapping, change detection, and outbreak trend analysis of coronavirus (COVID-19) in Iran (days between February 19 and June 14, 2020).* International Journal of Infectious Diseases, 2020. **98**: p. 90-108.

97. Ahasan, R., et al., *Applications of GIS and geospatial analyses in COVID-19 research: A systematic review.* F1000Research, 2020. **9**.

98. Alassafi, M.O., M. Jarrah, and R. Alotaibi, *Time series predicting of COVID-19 based on deep learning.* Neurocomputing, 2022. **468**: p. 335-344.

99. Santosh, K., *COVID-19 prediction models and unexploited data.* Journal of medical systems, 2020. **44**(9): p. 170.

100. Hira, S., A. Bai, and S. Hira, *An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images.* Applied Intelligence, 2021. **51**: p. 2864-2889.

101. Apostolopoulos, I.D., S.I. Aznaouridis, and M.A. Tzani, *Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases.* Journal of Medical and Biological Engineering, 2020. **40**: p. 462-469.

102. Shahid, F., A. Zameer, and M. Muneeb, *Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM.* Chaos, Solitons & Fractals, 2020. **140**: p. 110212.

103. Roy, S., G.S. Bhunia, and P.K. Shit, *Spatial prediction of COVID-19 epidemic using ARIMA techniques in India.* Modeling earth systems and environment, 2021. **7**: p. 1385-1391.

104. Kumar, S., et al., *Analysis of COVID-19 outbreak using GIS and SEIR model*, in *Fractional Order Systems and Applications in Engineering*. 2023, Elsevier. p. 215-225.

105. Razavi-Termeh, S.V., et al., *Covid-19 risk mapping with considering socio-economic criteria using machine learning algorithms.* International journal of

environmental research and public health, 2021. **18**(18): p. 9657.

106. McCoy, D., et al., *Ensemble machine learning of factors influencing COVID-19 across US counties.* Scientific reports, 2021. **11**(1): p. 11777.

107. Ma, R., et al., *The prediction and analysis of COVID-19 epidemic trend by combining LSTM and Markov method.* Scientific Reports, 2021. **11**(1): p. 17421.

108. Yahya, B.M., F.S. Yahya, and R.G. Thannoun, *COVID-19 prediction analysis using artificial intelligence procedures and GIS spatial analyst: a case study for Iraq.* Applied Geomatics, 2021. **13**: p. 481-491.

109. Khan, F.M., et al., *Projecting the criticality of COVID-19 transmission in India using GIS and machine learning methods.* Journal of Safety Science and Resilience, 2021. **2**(2): p. 50-62.

110. Niu, B., et al., *Epidemic analysis of COVID-19 in Italy based on spatiotemporal geographic information and Google Trends.* Transboundary and Emerging Diseases, 2021. **68**(4): p. 2384-2400.

111. Munkhdalai, L., et al., *Mixture of activation functions with extended min-max normalization for forex market prediction.* IEEE Access, 2019. **7**: p. 183680-183691.

112. Kim, H.-J., J.-W. Baek, and K. Chung, *Associative knowledge graph using fuzzy clustering and Min-Max normalization in video contents.* IEEE Access, 2021. **9**: p. 74802-74816.

113. Baldi, P. and P.J. Sadowski, *Understanding dropout.* Advances in neural information processing systems, 2013. **26**.

114. Korkalainen, H., et al., *Detailed assessment of sleep architecture with deep learning and shorter epoch-to-epoch duration reveals sleep fragmentation of patients with obstructive sleep apnea.* IEEE journal of biomedical and health informatics, 2020. **25**(7): p. 2567-2574.

115. Liu, Z.-H., et al., *A regularized LSTM method for predicting remaining useful life of rolling bearings.* International Journal of Automation and Computing, 2021. **18**: p. 581-593.

116. Hawkins, D.M., *The problem of overfitting.* Journal of chemical information and

computer sciences, 2004. **44**(1): p. 1-12.

117. Phaisangittisagul, E. *An analysis of the regularization between L2 and dropout in single hidden layer neural network*. in *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*. 2016. IEEE.
118. Alrifaey, M., W.H. Lim, and C.K. Ang, *A novel deep learning framework based RNN-SAE for fault detection of electrical gas generator.* IEEE Access, 2021. **9**: p. 21433-21442.
119. Tato, A. and R. Nkambou, *Improving adam optimizer.* 2018.
120. Singarimbun, R.N., E.B. Nababan, and O.S. Sitompul. *Adaptive moment estimation to minimize square error in backpropagation algorithm*. in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*. 2019. IEEE.
121. Bowling, M. and M. Veloso, *Multiagent learning using a variable learning rate.* Artificial intelligence, 2002. **136**(2): p. 215-250.
122. Gori, M. and A. Tesi, *On the problem of local minima in backpropagation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(1): p. 76-86.
123. Usharani, B., *ILF-LSTM: Enhanced loss function in LSTM to predict the sea surface temperature.* Soft Computing, 2023. **27**(18): p. 13129-13141.
124. Karevan, Z. and J.A. Suykens, *Transductive LSTM for time-series prediction: An application to weather forecasting.* Neural Networks, 2020. **125**: p. 1-9.
125. Wang, Z., et al., *LSTM-convolutional-BLSTM encoder-decoder network for minimum mean-square error approach to speech enhancement.* Applied Acoustics, 2021. **172**: p. 107647.
126. Liu, X., J. Zhou, and H. Qian, *Short-term wind power forecasting by stacked recurrent neural networks with parametric sine activation function.* Electric Power Systems Research, 2021. **192**: p. 107011.
127. You, Y., et al. *Large-batch training for LSTM and beyond*. in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2019.
128. Park, J., D. Yi, and S. Ji, *A novel learning rate schedule in optimization for neural networks and it's convergence.* Symmetry, 2020. **12**(4): p. 660.

129. Santiago, M.A., G. Cisneros, and E. Bernués, *An MLP neural net with L1 and L2 regularizers for real conditions of deblurring.* EURASIP Journal on Advances in Signal Processing, 2010. **2010**: p. 1-18.

130. Salem, H., et al., *Predictive modelling for solar power-driven hybrid desalination system using artificial neural network regression with Adam optimization.* Desalination, 2022. **522**: p. 115411.

131. Bahri, A., et al., *Remote sensing image classification via improved cross-entropy loss and transfer learning strategy based on deep convolutional neural networks.* IEEE Geoscience and Remote Sensing Letters, 2019. **17**(6): p. 1087-1091.

132. Bai, Y. *RELU-function and derived function review*. in *SHS Web of Conferences*. 2022. EDP Sciences.

133. Sekerli, M., et al., *Estimating action potential thresholds from neuronal time-series: new metrics and evaluation of methodologies.* IEEE Transactions on Biomedical Engineering, 2004. **51**(9): p. 1665-1672.

134. Rjoob, K., et al., *Machine learning and the electrocardiogram over two decades: Time series and meta-analysis of the algorithms, evaluation metrics and applications.* Artificial Intelligence in Medicine, 2022: p. 102381.

135. Chai, T. and R.R. Draxler, *Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature.* Geoscientific model development, 2014. **7**(3): p. 1247-1250.

136. Chai, T. and R.R. Draxler, *Root mean square error (RMSE) or mean absolute error (MAE).* Geoscientific model development discussions, 2014. **7**(1): p. 1525-1534.